Methods and models for the quantitative analysis of crowd brainstorming

by

Filip Richard Krynicki

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Microtask marketplaces provide shortcuts for automating tasks that are otherwise intractable for computers. Creative tasks fall squarely within this definition, and microtask marketplaces have been heavily leveraged to this end [6, 17, 31, 35, 38, 39, 46, 33]. Brainstorming is often an implicit component of these solutions. This thesis provides the first foundational study of brainstorming in microtask marketplaces, aimed at solving the open problems in brainstorming task design to make this process more accessible and effective. This is achieved by establishing techniques for coding brainstorming data at scale, models for quantifying desirable outcomes of brainstorming, and a qualitative deconstruction of brainstorming strategies employed in this environment.

Idea forests are introduced as a data structure to enable the disambiguation of ideas in large corpuses, providing natural measures of two metrics of primary interest in brainstorming research: quantity and novelty. They are constructed via a tree-traversal algorithm, restricting the subset of the corpus which the coder must be aware of when making decisions. A simulation approach is introduced to assess the validity of hypothesis outcomes derived from idea forest metrics.

The introduction of idea forests enables the core contribution of this thesis, a set of quantitative models for brainstorming outcomes. This thesis extracts several actionable conclusions from the parameters of these models: the rate of unique idea generation is subject to decay over time; individuals have a significant effect on the rate of idea generation, with productive workers generating dozens more unique ideas; and individuals generate their most novel ideas late in a brainstorming session, after the first 18 responses. Furthermore, a replication of findings by Nijstad and Stroebe [47] is conducted, finding that workers take more time to generate ideas when changing semantic categories and are more likely to remain within a category than expected by chance.

Finally, a taxonomy of strategies employed by brainstormers is presented. In particular, this thesis discusses the phenomena of scoping brainstorming problems, providing partial solutions, and riffing on previous solutions.

Acknowledgments

While I am obligated to state that I am the sole author of this thesis, this is true only in the most superficial sense. The ideas and words within are the product of many individuals' contributions, for which I have had the good luck to be the receiver.

First I must thank Michael Terry, my supervisor. His dynamism in my undergraduate lectures was a key component in the transformation my education from credentialism to curiosity, excitement and optimism. He supplied the continual positivity necessary for the completion of this thesis. I must also thank him for encouraging deep knowledge in daunting domains. Thanks to Parmit Chilana, Dan Lizotte, and Krzysztof Gajos for being readers on my thesis, but more importantly for their contributions to my education in classrooms, seminars and meetings.

I must also thank my friends and co-researchers at the University of Waterloo: Ben Lafreniere, Adam Fourney, Matt Kay, Jeff Avery, Valerie Sugarman, Tyler Szepesi, Ben Cassell, William Saunders, Matei Nagalescu, Jaime Ruiz, Yuexing Luo, and Yi Ren. You have alternately acted as friends, supervisors, confidants, and role models. In particular, thank you for taking my acerbic humour in good turn. I also thank Pao Siangliulue for her collaboration on this project. Furthermore, credit must go to Jamie Murdoch, Chris Zheng, and Huang Zijue, who assisted with this work as undergraduate researcher assistants.

Outside the academic sphere, I was especially assisted by members of the Stan users mailing list (stan-users@googlegroups.com) in the construction and debugging of models. In particular, Bob Carpenter was always available for assistance and education.

My family deserves special thanks. My parents Richard and Gabe provide overwhelming support and unflappable good nature. Maria's trust in me as her big brother is one of my most prized possessions.

Finally, and most importantly, I thank Ban. You have provided everything that supports this work: confidence, comfort, relaxation, perspective, and humour, to name a few. Thank you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis serves as a foundational study of the fundamental properties of brainstorming in microtask marketplaces. Already, researchers and practitioners are using microtask marketplaces to solve problems requiring creativity. However, despite this natural fit of problem domain and platform, prior work has yet to produce design guidelines and best practices for crowd idea generation tasks. This thesis addresses some of the most basic questions of brainstorming task design for microtask marketplaces:

- How many ideas should be requested from each worker?
- How many workers should be asked for ideas?
- What is the evaluation criteria for responses?
- How do workers brainstorm differently in microtask marketplaces than other environments?

These questions are addressed across three major contributions to this thesis:

1. Evaluation criteria are provided in the creation of a data structure, the *idea forest*, which enables quantifying brainstorming outputs at scales enabled by microtask marketplaces.
2. The brainstorming outcomes of quantity and novelty are modeled to derive guidelines for selecting workers and quantities of work.

3. Brainstormers responses' are qualitatively examined to understand what processes they apply in this environment.

In this chapter, I will motivate this thesis, enumerate its contributions, and provide an overview of thesis structure.

1.1 Motivation

Crowdsourcing is an increasingly popular solution to problems that are difficult to automate. One of the more popular domains for crowdsourcing is *microtask marketplaces*, web marketplaces such as Amazon's Mechanical Turk [1] where individuals can accept small tasks for monetary reward. Researchers and practitioners have utilized microtask marketplaces for article or review writing [6, 17, 31, 35, 38, 39], design [46], art [33], image or language recognition [64, 48, 39], and human subjects experiments [32], to name a few. Writing, design and art are particularly appealing crowdsourcing tasks because they leverage human intelligence for *creativity*, a domain which is as of yet impossible to explore without human input.

For example, Zhang et al. [72] leveraged human creativity in their Mobi system to generate ideas for activities to be enjoyed on a vacation. Mobi's use of individuals to generate lists of activities is an example of a wider trend. Often, a prerequisite to creative work in crowd environments is *idea generation*, the process of developing ideas in response to a specific prompt. In Mobi, crowd workers must generate ideas that meet a prompt containing the constraints of a vacation. As another example, Nickerson and Sakamoto [46] asked crowd workers to mix designs for chairs — a prerequisite of this design process was having workers generate their initial designs.

Brainstorming is a classic solution the problem of idea generation. Brainstorming was introduced by Osborn in 1957 [49] and is primarily defined by the principle of deferred judgment: ideas should not be subject to evaluation during the generation phase, but instead be used as prompts for further generation. In popular culture, brainstorming is often presented as an explicitly group process, but the principles of brainstorming can be equally applied individually. The process of individual ideation following the principles of brainstorming is known as *nominal brainstorming*. Nominal brainstorming has been found to perform better than group brainstorming [62, 55], and to not be subject to many of the problematic social pressures of group brainstorming [45, 47, 28].

[1] http://www.mturk.com

Nominal brainstorming is a natural fit to the problems of idea generation in microtask marketplaces. In particular, it describes individual ideation separated temporally and spatially; in the microtask marketplace environment, participants work on their own schedules and cannot easily communicate. Little et al. [40] were the first to explicitly consider brainstorming in the context of microtask marketplaces, and found that a nominal brainstorming process (referred to in their work as iterative) produced the highest quality responses. Furthermore, many idea generation steps in crowdsourced systems already implicitly support the core brainstorming principle of deferred judgment. For example, in Mobi ideas are voted on after generation, and Little et al. have an explicit evaluation step later in their brainstorming workflow.

Despite the application of brainstorming principles, there has been no study of the fundamental properties of brainstorming in microtask marketplaces. For example, generally idea generation tasks ask for an arbitrary number of responses, without considering how this affects outcomes of interest. Furthermore, while quantitative *models* of brainstorming exist for traditional brainstorming processes, these models are not easily applied in the context of a microtask marketplace. First, most of these models [10, 27, 47] emphasize the social nature of traditional group brainstorming, and aim to disambiguate social factors. Second, there are qualitative differences in the environment provided by a microtask marketplace, including variance of effort evidence between workers [6], the factor of monetary compensation that may lead to gaming of the system [32], the lack of spatial and temporal co-location, and the ability to economically collect ideas at massive scale. Models and baselines of brainstorming performance in microtask marketplaces are a first step towards creating systems that maximize creative output in the idea generation phases of creative task solving.

It is important to state plainly what my expectations are for brainstorming in microtask marketplaces. Obviously, crowd brainstormers are not expected to generate useful ideas for the purposes of curing diseases or proving $P \neq NP$. However, there are many tasks for which it can be expected that members of crowdsourcing communities have sufficient expertise. For example, participants could provide design feedback for a system that they can be expected to be familiar with; one of the questions used to establish a data set for this thesis asked workers on Mechanical Turk to brainstorm features and functionality for a mobile companion app. Similarly, crowds can be expected to have sufficient expertise to consider problems that people face in their everyday lives regarding products, problems, social situations, and so on. Ultimately, I see crowd brainstorming being deployed in two avenues: by individuals seeking to solve everyday problems, and by organizations looking to leverage the general wisdom of the crowd to provide better goods or services. Interest in the former scenario is demonstrated by advice requesting on websites such as Yahoo

Answers[2], and interest in the latter by the omnipresent feedback and survey requests on organization websites.

Researchers, as previously established, already make use of microtask marketplaces in a brainstorming context [32, 39, 35, 72]. Models of brainstorming behaviour would benefit researchers, by providing rules of thumb for maximizing brainstorming outputs as components of larger creativity workflows, and a framework for comparing future designs and interventions. These benefits are not only applicable to HCI and design communities. For example, an understanding of how crowdsourced nominal brainstorming compares to offline brainstorming could allow social science practitioners to make use of microtask marketplaces for expedited studies, similar to the contributions of Kittur et al. [32] in replicating HCI studies on Mechanical Turk.

Practitioners also stand to gain from improved understanding of brainstorming in microtask marketplaces. Businesses and individuals that rely on these marketplaces to perform automated creativity tasks are benefited by models that allow them to optimize brainstorming tasks for certain outcomes. For example, practitioners may wish to generate a large volume or ideas, generate more original ideas, generate in-depth or broad ideas, or predict abilities of a potential worker before soliciting the majority of a large and expensive body of work. In an ideal world, it would be possible for practitioners to provide a question, a budget, and perhaps a few evaluation criteria, and an algorithm would automatically generate tasks, solicit workers, and combine their responses ordered by quality.

Microtask marketplaces are already being used for creative work and brainstorming in research and business. Crowd workers are already engaging in creative work on these platforms. Best practices for comparison of brainstorming strategies, maximization of brainstorming outcomes, and design of brainstorming tasks are as of yet undefined. This work provides a first step towards filling this hole in our ability as a community to effectively apply mass human resources to idea generation.

1.2 Thesis statement

This thesis addresses the lack of foundational knowledge of brainstorming in microtask marketplaces. Earlier in this chapter I identified the design problems of choosing the number of workers, choosing the number of responses, creating evaluation criteria, and understanding how workers brainstormed. This thesis tackles these problems as they manifest in the following research questions:

[2]http://answers.yahoo.com

1. How can brainstorming responses gathered at large scale in a microtask marketplace be coded and organized?
2. What are accurate models for outcomes of interest in creative work, particularly rate of unique idea generation and the novelty of ideas?
3. How much do individuals vary in the rate at which they produce new ideas?
4. When do individuals generate their most novel ideas?
5. Are the performance and properties of brainstorming in a microtask marketplace comparable to traditional brainstorming?
6. How do individuals brainstorm in a microtask marketplace; what strategies do they employ to generate ideas?

1.2.1 Contributions

In this thesis, I make several contributions:

Models of microtask marketplace brainstorming. Two primary outcomes of interest for idea generation tasks are the quantity of unique ideas and the novelty of those ideas. Three models of the rate of idea generation are introduced. The first of these models verifies that participants generate ideas non-linearly; they decay over time. The second rate model encodes this finding to better describe the real-world idea generation process. The final rate model leverages the increased descriptiveness of the second model to show that participants have a significant effect on brainstorming outcomes, with productive participants generating dozens more unique ideas. Another model describes how the novelty of ideas changes over the course of a brainstorming session, and is applied to empirically derive the point at which workers begin to produce their most novel ideas: after the first 18 responses.

Taxonomy of brainstorming strategies. Not all responses to a brainstorming prompt are equal. This thesis presents a taxonomy of brainstorming strategies that result in qualitatively different kinds of responses. Three kinds of strategies are identified. Problem scoping strategies transform a brainstorming prompt into one that is easier to generate ideas for, such that the ideas still solve the original prompt. Riffing strategies manipulate or combine previous responses to generate new responses. Partial solutions provide information that may be useful, but does not fully specify a solution to the brainstorming problem. Strategies present an alternative angle for understanding brainstorming. For

example, problem scoping strategies are particularly common, suggesting that participants rarely leverage all the details of a problem in their solution, which may impact solution quality. Strategies could also act as potentially easy-to-detect signs of behaviour to prompt interventions in online brainstorming tasks.

Methodology for microtask brainstorming at scale. A formative study of creative tasks in a new environment requires collection of data at scale. This presents unique collection, coding, and verification challenges. *Idea forests* are introduced as a structure for encoding hierarchical generalization relationships between ideas. An algorithm for producing idea forests in a semi-automated fashion is presented, as well as a supporting tool for human coders. Constraints on the properties of idea forests are defined, and a novel simulation-based approach to validation of these constraints is introduced. Finally, this methodology is grounded in the existence of a 10,000-response corpus of brainstorming responses collected for the purposes of this thesis.

Replication. The SIAM model of brainstorming activity by Nijstad and Stroebe [47] makes predictions regarding the impact of *category switching* on the time it takes an individual to generate ideas. These predictions are tested and found to hold in the context of microtask market brainstorming. This provides preliminary evidence that the process of nominal brainstorming in an online environment performs similarly to traditional nominal brainstorming.

1.3 Organization

Chapter 2 of this thesis is a survey of prior work in the domains of information retrieval and machine learning, brainstorming, crowdsourcing and crowd creativity.

Chapter 3 describes the study conducted to collect a corpus of brainstorming responses. It also introduces quantity and novelty as metrics of interest in brainstorming tasks, and describes how they are derived from a corpus using a novel organization of brainstorming data called an *idea forest*.

Chapter 4 describes the methodological contributions of this work. The process for constructing an idea forest from a corpus of brainstorming responses is described. Validity tests are described and the results are given for the gathered corpora.

Chapter 5 includes the specifications for models of rate of idea generation and idea novelty. It describes the decision-making process in reaching these models definitions, comparisons between models, and describes the implications of these models in the context

of crowd brainstorming. It is found that the rate of new ideas decays over time; that workers are a significant cause of variation, with the most productive workers generating dozens more unique ideas; and that workers generate their most novel ideas after the first 18 responses.

Chapter 6 describes the qualitative coding process that I used to extract meaningful strategies from the brainstorming corpus. Definitions are given for the brainstorming strategies, as well as descriptive statistics of their prevalence in one of the data sets gathered. Potential applications for this strategy taxonomy are discussed.

Chapter 7 discusses of the pertinent high-level ambitions for crowd brainstorming research. These goals are addressed in the context of this study in the form of insights gained which were not explicitly tested in this thesis.

Chapter 8 provides a roadmap of pertinent future work to move towards the goal of effective idea generation in microtask marketplaces.

Chapter 9 summarizes the above contributions.

Chapter 2

Related work

2.1 Overview

This chapter provides a summary of prior work on brainstorming and crowd creativity that is particularly applicable to this work. Brainstorming is introduced as well as the development of brainstorming in electronic contexts. The problems inherent to measuring brainstorming, idea disambiguation and creativity scoring, are discussed in the context of previous solutions. Previous models of brainstorming are briefly described. Finally, this chapter describes crowdsourcing research and in particular surveys work in which the crowd is leveraged for creativity.

2.2 Brainstorming

In 1957, Osborn [49] introduced brainstorming, a technique to increase the productivity of idea generation in groups. In Osborn's description, brainstorming is a single component of the larger creative decision making process [28], the goal of which is to generate a list of ideas that can later be evaluated. This perspective informs the treatment of brainstorming within this work: brainstorming as a component of larger workflows for solving creative problems using microtask marketplaces.

Osborn's technique is defined by four rules of priority and group conduct, facilitated by a trained mediator (quoted from Isaksen [28]):

1. Criticism is ruled out. Adverse judgments must be withheld until later.

2. Freewheeling is welcomed. The wilder the idea, the better; it is easier to tame down than to think up.

3. Quantity is wanted. The greater the number of ideas, the greater likelihood of useful ideas.

4. Combination and improvement are sought. In addition to contributing ideas of their own, participants should suggest how the ideas of others can be turned into better ideas.

These rules take two forms. The first, second, and third rules encourage participants to defer judgment. Even if an idea is poor, consideration of it may provoke the generation of a better idea subsequently. The last rule aims to maximize the quantity of ideas generated. Osborn's rules are most commonly cited as the central definition of the brainstorming process. However, Osborn's definition includes several other measures to improve the success of brainstorming groups [28]:

- A brainstorming group should be accompanied by a facilitator; participants should have some expertise in the problem domain; recording should be done by a person [or technology] distinct from the individuals engaged in brainstorming.
- Individual ideation should be done before a group session as well as after.
- Participants should receive prior training in the brainstorming technique.

In his review of brainstorming research, Isaksen notes that most of these extended guidelines are not taken in to consideration in studies after Osborne's own [28]. In the interest of comparison and compatibility with the qualities of microtask marketplaces, this work focuses primarily on the four rules as a basis for brainstorming task design.

Zagona et al. [71] identify several kinds of creative problems: those which aim to *explain* a phenomena, those which aim to *predict* future consequences, and those which aim to *invent* a new set of conditions which will precipitate some outcome. Zagona also examines decomposing brainstorming questions along different axes, such as "reality" or groundedness. In one study [41], the "reality" or groundedness of a question had no effect on solution outcomes. In another [50], unreal questions generated more ideas. In general, differences between questions have been found to be a large source of variance in the quantity of ideas generated. It remains an open research question to identify and extract the properties of brainstorming questions, and determine to what extent these have a causal effect on outcomes. While this thesis does not directly address this question, it provides further evidence for differences in quantity outcomes between brainstorming questions.

2.3 Brainstorming measures

To model brainstorming outcomes, those outcomes must be quantified. There are two steps to quantification in most brainstorming work. First, ideas must be disambiguated so that multiple ideas that refer to the same semantic solution do not contribute to the overall quantity or quality of a participant or group. Second, each semantic solution may be evaluated for a metric of interest. Normally, this metric of interest is the *creativity* of an idea.

2.3.1 Idea disambiguation

The idea disambiguation problem has been a traditional prerequisite to brainstorming research. Given a set of natural language responses to a brainstorming prompt, idea disambiguation is the problem of determining which responses refer to an identical semantic idea. A simple example of the disambiguation problem is to determine that "paperweight" and "hold down paper" refer to semantically identical responses to the question "what to do with my favourite rock".

Taylor et al. [62] briefly describe a technique in which responses are decomposed into *steps*, complete with categories and subcategories thereof, but do not describe how categories were identified nor the criteria for determining to which category an idea belongs. Furthermore, the designation of steps only applies to responses which explicitly encode a plan of action. In Chapter 6, it will be shown that many brainstorming responses do not provide a plan.

In the Bouchard and Hare experiments [8], ideas were disambiguated by judges. A set of rules were employed: discard ideas that are too general, discard misunderstandings, and count lists of examples as single ideas. This technique was also utilized by Gallupe et al. [22] and Pinsonneault et al. [55]. This technique has several weaknesses. First, because number of ideas was the only measure of interest, the resulting value is a single scalar for the total count. It is unclear if the resulting disambiguated set is similar between judges. Second, the technique does not specify the granularity for disambiguating between ideas. In the course of this thesis work, I found that granularity needed to be considered as an explicit property of the coding process to produce agreement between coders.

Diehl and Stroebe [16] apply a technique in which judges linearly examine groups of ideas and remove any that had been proposed before. This technique and those above are precluded from application in this thesis by the scale of data collected. Microtask marketplaces provide explicit advantages of scale, and in this thesis thousands of responses

are gathered, making it difficult for coders to maintain a mental record of previous ideas without assistance.

Dennis and Valecich [15] instead coded ideas at a categorical scale. Using their example, the idea "Professor Jones" would be coded as "faculty". This coding method throws away significant information, and further assumes a strictly two-level hierarchy, in which all categories encompass a similar degree of generality.

2.3.2 Measuring creativity

Creativity is the most desirable measure of interest for a brainstorming task. Zagona et al. [71] provide an early survey of creativity measures as applied to group creative problem solving. Among variables contributing to creativity measures are productivity, ingenuity, novelty, combination, practicality/appropriateness, variety, and quality [19, 59]. These sub-measures provide a daunting number of ingredients from which to mix a creativity cocktail for any single work.

Further complicating matters is correlation between measures of creativity. Diehl and Stroebe [16] found that number of responses generated and quality of responses were highly correlated. This finding has been replicated many times [52, 51, 59], including in the domain of electronic brainstorming [9].

A common method of resolving the creativity measurement problem is to select a small subset of the above measures, have judges score them on ordinal scales, with the outcome of interest some function of those scores. Lewis et al. utilized expert raters to score responses to an idea generation task on 5 point scales for originality, feasibility, elaboration, and flexibility [37]. Marsh et al. had raters score drawings of creatures by the number of distinct features of the creature they could identify s[42]. Meadow and Parnes [52] used this method for generating a creativity score based on uniqueness and value. They employed the common practice of simplifying the resulting score by assigning a binary creativity score based on thresholds across sub-measures. Diehl and Stroebe [16] used a similar measure combining originality and feasibility. Yu and Nickerson[69] also employed a combination score in which crowd workers rated for originality and practicality on a 5-point scale, such that ideas that exceeded 4 on both measures were considered creative. In the course of this thesis, these score-based measurements were found to suffer in inter-rater reliability at the large scale of data employed. For example, it is difficult for judges to assess uniqueness of a single idea in a data set of thousands.

Soukhoroukova et al. [60] propose *idea markets*, infrastructures for evaluating the value of ideas by trading them in a stock market and examining their posterior trade

value. Soukhoroukova applies this technique to evaluating product concepts for four MP3 players, and found that the price of an idea on the market correlated with the predicted market shares. It is unclear that this technique could generalize to very large data sets, where trading participants could not be expected to build up a familiarity with every idea.

An alternative is to introduce a deterministic, quantitative measure of creativity (or a sub-component of creativity). In their paper examining the phenomenon of *design fixation* among those involved in a creative task, Jansson and Smith [29] introduced a measure for novelty that follows from any solution for idea disambiguation, the *o-score*. In this thesis, the o-score is employed to compare the novelty of individual ideas. Given a disambiguated group of creative responses, the o-score is given by the following equation:

$$1 - \frac{\text{\# of instances of semantically identical idea}}{\text{total \# of ideas in data set}}$$

A strong advantage of the o-score is that it is naturally defined in terms of idea disambiguation. It encompasses only one component commonly used in creativity measures, namely novelty (alternatively originality or uniqueness). Novelty, while not a sufficient metric for creativity alone, is a critical component of virtually all creativity measures. Furthermore, o-score is easily compared between data sets and judges provided the idea disambiguation step is reliable.

2.4 Brainstorming groups

While the common understanding of brainstorming refers to group ideation, Taylor at al [62] contest the effectiveness of groups in the brainstorming process. They compared regular brainstorming groups as expected in the Osborn process to *nominal groups*: groups of individuals who generate ideas separately that are later combined into a single pool, with duplicate ideas removed. Individuals participating in nominal brainstorming are still given the rules of brainstorming prior to the exercise. They found that participants brainstorming in nominal groups generated more ideas, more unique ideas, and higher quality ideas. Bouchard and Hare extended this finding to account for the size of brainstorming groups, finding that large groups produced more ideas than small groups, and that nominal groups produced more ideas than regular groups [8]. Furthermore, the rate of idea generation for nominal groups grew linearly as a function of the number of members in the group. However, the maximum group size explored in this study was nine, far fewer than might be expected to participate in a brainstorming task on a microtask marketplace. In the course of this study, I found this linear growth finding did not apply.

Three major reasons for the productivity loss in real group brainstorming are commonly cited [16]. *Production blocking* is an artifact of the bandwidth for spoken communication in a group brainstorming session: only one person can speak at once. When and if a member finds a lull in the conversation to contribute, they have forgotten ideas. *Evaluation apprehension* is the fear of judgment by other members of the brainstorming session, or facilitators. This difficulty persists despite the "defer judgment" rule of brainstorming [12]. *Free riding* occurs when participants feel less pressure to generate ideas because others within the group are being productive. A fourth reason introduced by Camacho and Paulus [11], *social matching*, occurs when members of a group who would normally have high productivity reduce their performance to that of less productive members, to force the less productive members into equal contribution.

To investigate these factors, Diehl and Stroebe [16] conducted several experiments, finding that evaluation apprehension and free riding had a significant effect on number and quality of produced ideas, but that production blocking represented the majority of productivity loss.

Despite these effects, there remains a romantic appeal to the idea that people working together could generate better ideas than those apart if the social impediments could be mitigated. Computer-mediated or electronic idea generation is one well-researched attempt to execute on this idea. Participants work individually at computer terminals, entering ideas which are propagated to the other group members' displays. In theory, computer mediation reduces the impact of evaluation apprehension and production blocking, as users can enter ideas anonymously and simultaneously. Furthermore, computer-mediated brainstorming is thought to stimulate idea generation because participants can be inspired by and improve upon the ideas they see that were generated by others. The results of computer-mediated idea generation research have been mixed. Dennis and Valecich [15, 63] found that intact computer-mediated groups generated more ideas than nominal groups, and large electronic groups generated more ideas than smaller as would be expected. However, anonymity had no effect on the rate of idea generation. Gallupe et al. [22] found that groups using electronic brainstorming generated more ideas and more high-quality ideas than real groups. From this, they inferred that electronic brainstorming reduced the effects of production blocking and evaluation apprehension, though they make no claims as to whether simultaneous entry or anonymity provides more significant benefits.

Pinsonneault et al. [55] conducted a survey of the Dennis and Valecich, Gallupe et al, and other electronic idea generation work, as well as their own further studies, and found that nominal brainstorming always performed as well as or better than electronic (both anonymized and non-anonymized) and verbal group brainstorming. This is contextualized by suggesting that process gains identified for electronic brainstorming were not as powerful

as expected, and that the differences found by Dennis and Valecich were a result of an unorthodox technique for creating simulated data for nominal groups. Four additional process losses unique to electronic idea generation are identified:

- The distraction effect of reading others' ideas.
- Attentional production blocking (a participant cannot benefit from others' idea while focused on generating their own).
- Striving for originality (attempting to not replicate work in the shared session).
- Cognitive complexity (the addition of tasks such as reading ideas and interpreting them).

Electronic brainstorming has potential to deliver on aspects of group brainstorming in a microtask marketplace. Furthermore, it is tempting, particularly in the design-rich field of HCI, to propose many and elaborate interventions which are theorized to improve performance. However, nominal brainstorming remains the state of the art, at least effective as both group and electronic group brainstorming. Furthermore, existing idea generation techniques in crowd research are nominal in nature [72, 40]. If this thesis is to present baseline models and outcomes for brainstorming, then, it must adhere to this state of the art, both for its historical effectiveness and its direct applicability to the environment of microtask marketplaces.

Beyond the distinction of nominal vs real groups, there have been attempts to isolate the non-social influences on brainstorming outcomes. Parnes [51] tested the hypothesis that better ideas were generated later in an idea generation session. They found that subjects, both trained and untrained in brainstorming methodology, generated more good ideas in the latter half of a brainstorming session. They suggest that this is due to the extended effort on the part of the participant, and to the phenomenon of *deferment* [26]. Deferment describes the capacity of a participant to put off the satisfaction of problem solving by not accepting early solutions. Creative problem solvers are unsatisfied with an early idea because they set a minimum standard on the quality of ideas which they will accept. Thus, creative idea generators will continue to build upon and improve early ideas to achieve higher quality later. This suggests a "burn-in" period for idea generation, and a minimum threshold number of ideas solicited, before which lower-quality ideas are expected. This phenomenon has direct implications for the design of brainstorming tasks, and this thesis will examine it in the context of microtask marketplaces.

2.5 Models of brainstorming

There are several previous examples of models of brainstorming, particularly emphasizing the impact of social factors. Brown and Paulus [10] introduced a model that accounted for production blocking and matching, which assumes that the rate of idea generation is subject to exponential decay. The models in this thesis consider the rate of decay explicitly. Haddou et al. [27] expanded on the model of Brown and Paulus to allow real-time, online model generation and future prediction, with the intend that it be applied in live group brainstorming sessions. However, this model is evaluated for feasibility only; its predictions are not tested on real brainstorming data. These models focus on the impact of social factors in brainstorming. In contrast, the models in this thesis examine the process of nominal brainstorming. Furthermore, the models in this thesis are applied to a large corpus of real brainstorming responses. This thesis will also examine the assumption of exponential decay for nominal groups explicitly.

Nijstad and Stroebe [47] introduced the SIAM (search for ideas in associative memory) model for idea generation. While their focus was on idea generation in a group setting, their model also described individual brainstorming to a degree beyond the models described above. Under the SIAM model, idea generation is an iterative process that alternates between the phases of *knowledge activation* and *idea production*. In the knowledge activation phase, brainstormers bring to mind an *image*, a bundle of concepts, features and associations in response to the query. In the idea production phase, the brainstormer explores variations within this image to produce ideas. When the brainstormer has exhausted the image to a sufficient degree, they return to the knowledge activation phase and select a new image. Subsequent activations may involve recollection of items generated from previous images, and images can overlap.

Nijstad and Stroebe make several hypotheses that follow from SIAM, two of which are examined in this work:

1. Hypothesis 1: Ideas are more likely followed by an idea from the same category (image) than could be explained by random chance.

2. Hypothesis 2: Generating ideas when switching categories (images) will take more time than generating ideas within categories. This follows from the extra knowledge activation step.

In contrast to the Nijstad and Stroebe cognitive model which is used to derive and test hypotheses, the models in this thesis are applied to make quantitative estimates of values of interest, such as the rate at which idea generation decays.

2.6 Crowdsourcing

Crowdsourcing is a technique for problem solving that has received significant attention. Its basic premise is to employ large numbers of workers, with few assumptions as to expertise or skill, to solve complex problems. One of the most well-known contexts for crowdsourcing is in microtask marketplaces such as Amazon's Mechanical Turk, in which workers are paid to complete Human Intelligence Tasks, or HITs. HITs are often discrete units of work that can be completed in a short time frame, which may be later combined to provide a full problem solution. These microtask marketplaces allow requesters to conveniently automate tasks that require human intelligence, provided they can adequately compensate workers. They have been applied to a multitude of problems, from estimating the nutritional content of a plate of food [48] to reducing the length of written documents [6].

Many tasks that are both desirable to automate and require human intelligence are too large to be manageable for a single worker. Thus, significant effort has been taken to facilitate the decomposing of problems and recomposing of solutions. Little at al [38] introduced Turkit, an infrastructure for decomposing problems in which the requester constructs a workflow and outputs are dynamically recomposed into inputs for future steps. Other systems rely on the crowd to perform task decomposition/recomposition [35], or provide HIT-based analogs to common computation frameworks such as map-reduce [31]. In the context of these systems, crowd brainstorming can be considered as a step in a workflow in which the input is a prompt and the output is a list of ideas to be evaluated in later steps. In this thesis, a custom system was used which behaved similarly to Turkit, with additional systems to meet experiment requirements. It is available under the name *turkflow* on the Python Package Index[1].

Worker variability is another domain of crowdsourcing research with high applicability to this thesis. In their Soylent project, which demonstrated the ability of the crowd to perform in-interface operations, Bernstein et al. [6] found high variance in the output of workers, classifying extreme outliers as *Lazy Turkers* and *Eager Beavers*. Similarly, Kittur et al. identified "gaming" of the system on Mechanical Turk [32] by workers attempting to maximize financial outcomes by providing poor quality results that meet the minimum requirements of requests. Gaming workers create responses that are empty of information, non-constructive, or copy-and-pasted from others. Kittur et al. found that invalid responses were often identifiable by a short time of generation.

This diversity in output presents challenges for crowdsourcing systems, particularly in the domain of brainstorming. It is important to examine the extent to which participants

[1] https://pypi.python.org/pypi/turkflow/

provide useless information and affect outcomes during idea generation tasks.

2.6.1 Crowdsourcing creative problems

Crowdsourcing allows automated solution generation for a class of creative problems that are intractable for computers. As a result, creative problems have received significant attention in crowdsourcing research. Aaron Koblin's art project "The Sheep Market" [33] involved many participants on Amazon's Mechanical Turk each drawing a sheep facing left, all of which were combined into a single art piece. Another well-known example is von Ahm and Davish's ESP game [64]. In this system, two participants provide labels for an image without seeing their partner's labels. For example, an outdoor image may result in labels such as *green*, *sky*, and *clouds*. Whenever a word has been provided by both participants, they receive points and the label is accepted as descriptive of the system. Systems like these show the viability of the crowd for idea generation, but do not explicitly examine the process.

Nickerson and Sakamoto [46] propose a system in which ideas are iteratively pruned and improved upon by the crowd. Yu and Nickerson [69] expand on this and utilize crowd for the creative process of design, having participants design chairs for children. Participants generated a set of ideas for chairs which were later iteratively combined and mutated by other participants in an evolutionary algorithm. Idea generation is a key first step for seeding the genetic pool of ideas, and the later iterative recombination steps recall the fourth rule of Osborn's brainstorming: *combination and improvement are sought*. Yu and Nickerson found the crowd effectively produced new features and modified existing ones. They show that the genetic process resulted in better ideas, but do not examine properties of the generation phase such as the number of features different individuals contributed.

Zhang et al. [72] created Mobi, a system for crowdsourcing trip itineraries. Mobi is an example of a *crowdware* paradigm in which workers can see an overview of the entire work-in-progress solution, and select smaller units of work to make progress. In one of these units workers propose vacation activities that meet a set of constraints. For example, they may require that a portion of activities are appropriate for young children. This is analogous to a brainstorming problem, and demonstrates the importance of basic idea generation models and techniques to complex crowd creativity systems. Although the aim of Mobi — generate an entire trip itinerary — could not be said to be a strict brainstorming problem (it involves organization and sorting problems as well), brainstorming is a critical component of the process. Zhang et al. do not comment on the quantity or quality of ideas generated in this step, nor do they discuss how participants' ideas changed and varied for those that submitted more than one activity.

In another example of brainstorming as a prerequisite to a larger creative task, Kittur et al's Crowdforge system [31] was evaluated in the context of crowdsourcing article writing. One step of the article writing workflow required participants to generate a list of attractions for a trip to New York City. A further example is Turkomatic [35]. The example workflows presented in Turkomatic include essay-writing, a step of which requires participants to generate a list of facts. These systems do not describe the design considerations in their brainstorming steps; the implicit assumption is that soliciting ideas in *any* way is sufficient to produce satisfactory work to feed into a later step in the system. This thesis challenges these assumptions by demonstrating different performance between workers, questions, and phases of idea generation.

Little et al. [40] explicitly consider brainstorming as a creative task of interest in microtask marketplaces. In their experiment, participants had to generate five potential names for a company. They compared subjective ratings for brainstormed ideas in iterative (brainstormers could see previous contributors' ideas) and parallel (nominal brainstorming) conditions. They found that participants in the parallel condition generated the highest-rated ideas, consistent with the expectation that nominal brainstorming performed at least as well as alternatives. This study does not vary properties of the brainstorming task, such as question or number of ideas solicited. It does, however, provide a first guideline for brainstorming in a crowd environment: that nominal brainstorming structures provide the best results. This thesis builds on this finding and examines the process of nominal brainstorming in microtask marketplaces in more detail.

Finally, crowdsourcing to solve creative problems has been applied in various commercial settings. Bayus performs an analysis of the Dell IdeaStorm community, in which users and customers contribute product ideas and improvements [5]. Bayus finds that ideas coming later in the ideation process are more likely to be selected for implementation, which corresponds to an assertion of quality. However, once individuals had an idea selected, they were less likely to generate another quality idea. Bayus argues that this is because individuals tailor their successive ideas to be more like earlier accepted ideas, which limits diversity. This suggests that feedback to ideators should be deferred, particularly positive feedback as it may reinforce fixation behaviour. Several platforms have also been developed to serve as general marketplaces for idea generation tasks in particular, including Idea Bounty[2], Innocentive[3] and ideaconnection[4]. The latter two services in particular focus on the domain of expert crowdsourcing, assigning more difficult problems to experts in appropriate fields.

[2]http://www.ideabounty.com/
[3]https://www.innocentive.com
[4]http://www.ideaconnection.com/

2.7 Open problems

This body of work contains several open problems which are addressed in this thesis. Existing measurement techniques for brainstorming outcomes have not been applied in contexts in which the scale of ideas reaches that enabled by microtask marketplaces. In the course of this work alone, a corpus of 10,000 responses is gathered. Models of brainstorming in the social sciences literature have thus far focused on the social or cognitive characteristics of brainstorming. There remains the need for more *mechanical* models of brainstorming, which describe desirable outcomes such as quantity and novelty and explicitly examine assumptions encoded in variables, such as decay rate, in existing models. While crowdsourcing work has demonstrated an interest in creativity and employed idea generation as a means to an end, there has been a failure to examine the mechanisms of idea generation in crowd environments and explicitly consider the design of these idea generation steps. Finally, none of the existing work has examined how production occurs in a nominal environment. Even without a group influencing an individual, that individual is likely to encounter difficulties and employ strategies in the solo generation of ideas.

Chapter 3

Study

3.1 Introduction

The aim of this thesis is to present a foundational study of the properties of brainstorming tasks in microtask marketplaces. Specifically, I aim to create models of brainstorming which encode the process of nominal brainstorming in a microtask marketplace. These models will be fit to a corpus of brainstorming responses to ensure they accurately describe the process of crowd brainstorming, and the resulting parameter distributions will be examined to respond to the open research questions identified in Chapters 1 and 2. To accomplish this, a corpus of 10,000 responses to four brainstorming prompts was collected on Amazon's Mechanical Turk. This corpus serves three purposes. First, the corpus is used to fit models of brainstorming processes. Second, a qualitative coding of a subset of the corpus is made to identify strategies of idea generation employed by workers. Finally, the corpus provides a labeled set against which future work in the automated prediction of brainstorming outcomes can be tested.

This chapter describes the study by which the corpus was collected, as well as how the study design was derived. Specifically, it addresses the problem of selecting brainstorming questions. The chapter closes by introducing the measures of quantity and novelty extracted from the corpus, and giving descriptive statistics of the corpus.

3.2 Data collection

My goal was to develop quantitative baselines of brainstorming performance in microtask marketplaces. Previous work has used brainstorming as a use case to examine other phenomena [39] or as a component of a larger brainstorming workflow [72]. Thus, there does not exist a significant corpus of responses to brainstorming questions posed in a microtask marketplace without additional constraints imposed by the larger workflow. To that end, I opted to create a corpus of responses to brainstorming questions in the most naive implementation of a brainstorming HIT.

In a traditional group brainstorming session, a group of people meet in person (ideally with a facilitator) and share ideas verbally. By contrast, in a nominal group brainstorming session, an equal number of people each separately record ideas (verbally, written, or digitally), which are later collected and combined. As discussed in Chapter 2, nominal brainstorming sessions have been shown to perform at least as well as real group brainstorming sessions, and often much better. Nominal brainstorming conveniently aligns with the expected setting of a microtask marketplace task: work is performed spatially and temporally separated from collaborators. Furthermore, in existing brainstorming tasks or task components on microtask marketplaces, participants generate ideas separately which are later pruned and combined [39, 69, 72]. Thus, I assert that nominal brainstorming is the naive implementation of brainstorming in a microtask marketplace. All the brainstorming tasks presented to workers in this thesis are designed within the constraints of nominal brainstorming.

I created a template for nominal brainstorming HITs on Mechanical Turk. Participants were asked to generate 5, 10, 25, 50, 75, 100, or as many as possible responses to a brainstorming question, and were compensated \$0.18, \$0.35, \$0.70, \$1.75, \$2.65, \$3.50 and \$1.75 respectively (approximately 3.5 cents per response, with the unlimited condition paying the same as the 50 condition). This spectrum of responses grew as the data collection period continued; as quality responses were received to the 25 condition, the 50, 75 and 100 conditions were added in turn to test the limits of participants' ideation abilities.

After initial pilots, the unlimited condition was dropped from experiments and data analysis. This was due to the poor response quality from Turkers, with at most 15 ideas given in the condition. This is likely a result of the strong motivation for even quality Turkers to maximize their rewards by minimizing time spent on tasks. This is similar to the gaming phenomenon [32], but notably in this case the Turkers were meeting the stated requirements of the HIT, if not the spirit.

All recruited participants were residents of the United States. This decision was made

with the intent that it would ensure a minimum level of English language comprehension and a relatively consistent cultural background across participants.

Participants examining the HIT were informed of the number of ideas requested and the compensation before accepting the HIT. However, the question was not assigned or displayed to the participant until the HIT was accepted and they had given consent to participate. HITs were implemented as simple HTML forms with a page for consent followed by pages for response entry and finally a feedback letter. Upon giving consent, a request was sent to a separately-hosted server which would populate the form with a brainstorming prompt. This was done to ensure equal distribution of participants across questions, to limit selection bias, and to prevent the same participant responding to the same question multiple times.

Upon accepting the HIT and giving consent, participants were given a paraphrased version of Osborn's four rules of brainstorming [49]. The rules were paraphrased to remove references to an explicit group of brainstormers:

1. There are no bad ideas. Don't criticize your choices.
2. Wild ideas and building off of old ideas are okay.
3. Quantity of ideas is prioritized.
4. Combinations of ideas count as new ideas.

An example of the HIT HTML form is demonstrated in Figure 3.1. Participants were also able to give any number of additional responses in a free text field, but no participants chose to do so. Once accepted, participants had 18 hours to complete the HIT. This was chosen after participants in pilots pointed out that the previous limit (ten minutes plus one minute per question) did not allow them to claim HITs and then complete them at a later time, a common practice.

Giving the rules of brainstorming is generally the minimum baseline achieved in research to meet Osborn's brainstorming definition [62, 16, 22, 28]. However, Isaksen [28] argues that the rules themselves are not sufficient to meet the brainstorming guidelines outlined in Osborne's original work. Specifically, Isaksen argues:

1. Brainstorming groups should be chosen based on the nature of the problem such that they have sufficient and similar expertise.

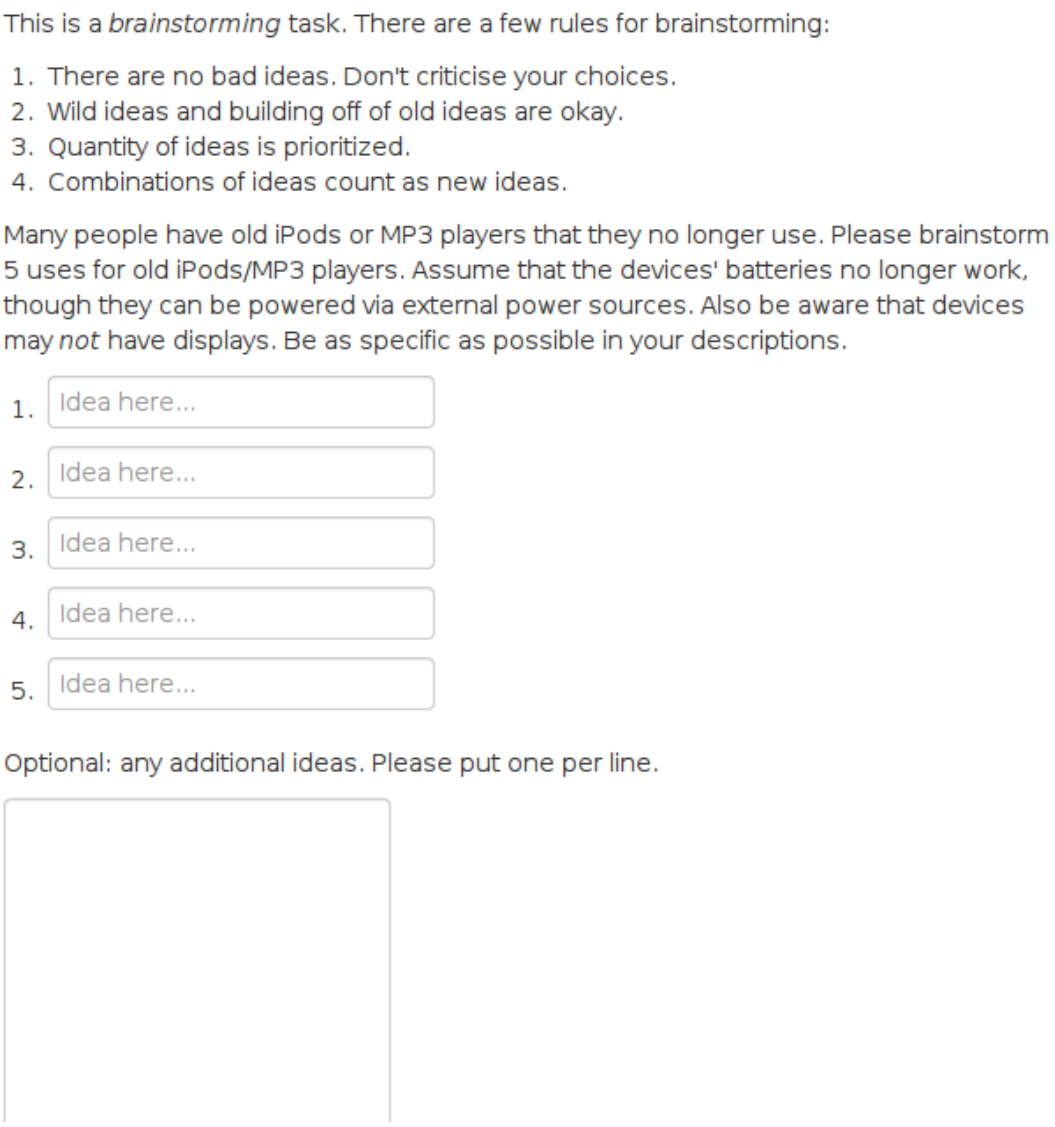


Figure 3.1: Sample task on Mechanical Turk

2. Individual ideation should occur prior to and following the brainstorming session.

3. Participants should be trained in brainstorming before participating in a session.

4. The session should have a trained mediator.

In the case of naive crowd brainstorming, it is unclear what the implementation of these recommendations would be. Given the restricted demographic information available for Turkers, it does not appear to be possible to meet the first guideline of assigning tasks based on domain expertise. The second guideline loses its meaning when all ideation is to occur individually, and furthermore participants in a microtask marketplace cannot be guaranteed to prepare for a task in advance or perform follow-up work. The third guideline, while achievable, would require extraordinary human resources and mitigate the primary advantages of a microtask marketplace — parallelization and efficiency. It is possible that video training could be employed, but this assumption would have to be explicitly tested for efficacy, and I leave it to future work. The final guideline poses similar difficulties to the second and third. It is unclear what the meaning of mediation of an individual is, and furthermore supplying mediators for each worker would dissolve the reasons for brainstorming in the crowd.

Finally, by limiting training to a statement of four rules, the responses in the brainstorming corpus established are better comparable to the majority of other studies with similar limited training (the state of the art that Isaksen laments). As a result of these difficulties and in the interest of comparability, I chose to maintain the simple statement of rules. That said, I recommend that these guidelines provide inspiration for potential future interventions. For example, an explicit training phase or having Turkers provide parallel mediation both prompt interesting research questions.

3.2.1 Question selection

I explored a variety of brainstorming questions in an iterative refinement process. Initially, I chose to use questions across the spectrum described by Zagona et al. [71]. The questions utilized were the "thumbs" problem (effects that would occur should you wake up with an additional thumb) the "mop" problem (alternative uses for a mop) and a weight problem (explaining influences on body weight). In initial pilots, the responses to these questions were rated by judges on a 5 point scale for creativity, realisticness, originality, and whether or not they were under-defined. Although this coding scheme had several weaknesses (it failed to achieve acceptable inter-rater reliability), it highlighted an overall poor quality of

the responses. On suspicion that the poor responses were due to the impractical nature of the questions, the questions were modified according to the first of Isaksen's above suggestions — that brainstormers have sufficient expertise for the problem domain:

- Please brainstorm N ways that Mechanical Turk could be improved for workers. Be as specific as possible in your descriptions.
- Please brainstorm N different public events that could be used to raise money for Alzheimer's research. Be as specific as possible in your descriptions.
- Many people have old iPods or MP3 players that they no longer use. Please brainstorm N uses for old iPods/MP3 players. Assume that the devices' batteries no longer work, though they can be powered via external power sources. Also be aware that devices may *not* have displays. Be as specific as possible in your descriptions.
- Imagine you are in a social setting and you have forgotten the name of somebody you know. Brainstorm N ways you could learn their name without directly asking them. Be as specific as possible in your descriptions.

Questions 3 and 4 immediately produced the desired improvements in subjective response quality. However, questions 1 and 2 continued to produce unsatisfactory responses. In particular, the Mechanical Turk question prompted workers to respond primarily that they should be paid more, and the charity question resulted in obvious answers such as bake sales and dinners. In order to remedy this, questions 3 and 4 were iteratively tested and refined until they excluded classes of responses deemed undesirable. This primarily manifested in additional constraints imposed on the questions with the intent to force workers to think outside the box. The revised questions 1 and 2 are printed below:

- The Electronic Frontier Foundation (EFF) is a nonprofit whose goal is to protect individual rights with respect to digital and online technologies. For example, the EFF has initiated a lawsuit against the US government to limit the degree to which the US surveils its citizens via secret NSA programs. If you are unfamiliar with the EFF and its goals, read about it on its website (`https://www.eff.org`) or via other online sources (such as Wikipedia). Brainstorm N *new* ways the EFF can raise funds and simultaneously increase awareness. Your ideas *must be different from their current methods*, which include donation pages, merchandise, web badges and banners, affiliate programs with Amazon and eBay, and donating things such as airmiles, cars, or stocks. See the full list of their current methods here: `https://www.eff.org/helpout`. Be as specific as possible in your responses.

- Mechanical Turk currently lacks a dedicated mobile app for performing HITs on smartphones (iPhone, Androids, etc.) or tablets (e.g., the iPad). Brainstorm N features for a mobile app to Mechanical Turk that would improve the worker's experience when performing HITs on mobile devices. Be as specific as possible in your responses.

I will refer to these four questions from here on in text, figures and tables as "turk", "charity", "iPod" and "forgot name".

3.2.2 Summary of data collected

For each individual response to a brainstorming question, the English text of the response was captured, as well as timestamps for the first and last activations of the form widget associated with that response. The duration of time to give a response was calculated as the difference of the latter two values.

Batches of responses were solicited over the course of several weeks. A breakdown of these responses is given in Table 3.1. Over all solicitations and conditions, 341 HITs with 280 unique Turkers were completed. 61 brainstorming HITs were completed by workers who had already completed at least one HIT. The data from repeat HITs *from the same worker with the same question* were removed from the corpus. This was done to avoid any learning effect of repeat exposure to the same brainstorming problem.

From Table 3.1, it is clear that significantly more responses were collected for the iPod question than the others. Initially, I chose to solicit the same number of brainstorming HITs for each question and number of responses requested condition, setting a lower bound at 10. However, during preliminary data analysis it became apparent that it was important that each number requested condition be reasonably represented on the response scale as well as in aggregate worker data. Further HITs were solicited for the iPod question, with the aim of a minimum of 400 individual responses in each condition. However, the 5 response condition was so unpopular that this minimum was not met, despite providing the same reward per question as the other HITs. This may indicate that workers prefer brainstorming HITs with high absolute value, or that these HITs are easier to find on Mechanical Turk. Furthermore, this lack of popularity was not uniform. The rate at which 5 response HITs were completed decreased over time. This suggests that it may be possible to exhaust the worker pool of Mechanical Turk with relatively few workers. In the case of this corpus, only 57 workers in total participated in the 5 response condition for the iPod question.

number of ideas requested	5	10	20	50	75	100
charity						
HIT responses	9	10	10	8	10	9
HITs excluding repeat workers	9	9	10	7	9	9
total ideas gathered	43	90	190	307	501	806
forgot name						
HIT responses	10	10	10	12	11	11
HITs excluding repeat workers	10	8	9	12	10	10
total ideas gathered	50	76	163	565	589	824
iPod						
HIT responses	59	49	23	10	10	10
HITs excluding repeat workers	57	39	21	9	10	10
total ideas gathered	293	372	413	450	634	855
turk						
HIT responses	11	10	11	10	9	9
HITs excluding repeat workers	11	9	11	9	9	7
total ideas gathered	55	84	215	402	620	699

Table 3.1: Result counts between conditions

3.3 Measures of quantity and novelty with idea forests

To model brainstorming, the outcomes of interest must be quantified. Brainstorming is creative work, and as described in Chapter 2, there are many methods for computing creativity scores based on judges. I found in the course of this work that that scale of the corpus made it impossible to achieve inter-rater reliability in these traditional measures. Instead, this thesis focuses on metrics of creativity can be extracted deterministically, namely *novelty* and *quantity* of ideas generated.

Quantity is derived by disambiguating *instances* (individuals' text responses to brainstorming prompts) into clusters of semantically identical *ideas*. Novelty is expressed in a metric called the *o-score*. The o-score is defined on the range 0-1, with a higher value indicating an idea that is more rare.

This section describes briefly how these metrics are computed from the corpus of brainstorming responses. This is achieved through a novel organization of brainstorming responses called an *idea forest*. An idea forest is a hierarchical representation of brainstorming responses that encodes relationships of semantic identity and semantic generalization.

3.3.1 Terminology and definitions

Throughout this thesis, I have referred to concepts such as *ideas* in an abstract sense, appealing to the reader's intuitive understanding of their meaning. Idea disambiguation requires a concrete definition of these terms and others. I will also introduce several supporting concepts: categories, runs, and instances.

Workers do not generate a set of responses in one atomic chunk. Instead, there is a temporal ordering to the responses given to a brainstorming task, and each response takes time to generate. I refer to the temporal organization of responses given by a single brainstorming participant as a *run*. If a brainstormer was asked to write down 10 solutions to a problem on a piece of paper, each to a line, then the paper, read top to bottom, would comprise the run. Each of the 10 lines is referred to as an *instance*, a single response to a single brainstorming problem by a single participant.

An *idea*, in contrast to an instance, refers to a *semantic* solution to the brainstorming prompt. The same idea may appear in multiple brainstorming runs. Instances and ideas can be visualized in a bipartite graph, as in Figure 3.2. Each instance node is connected to exactly one idea node, while each idea is connected to one or more ideas (a many-to-one relationship). Rather than consider instance nodes explicitly, they can be encoded as a

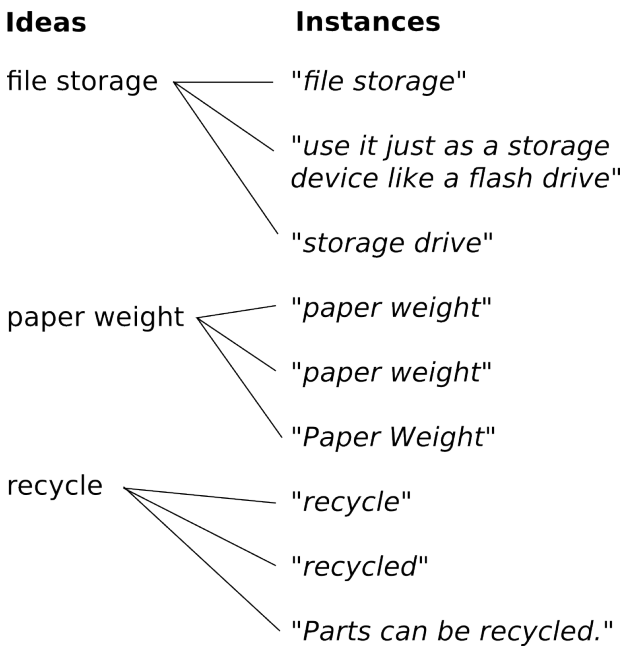


Figure 3.2: Ideas and instances have a one-to-many relationship. Instances are real text responses given by participants, while ideas have coder-supplied labels.

property of the idea node. I refer to this as the idea *mass*, the number of instances associated with an idea. All instances joined by an idea are semantically equivalent, with allowances for different phrasing. Consider the following example instances given in response to the iPod question:

1. Storage container
2. Small storage box
3. Coin storage
4. Travel jewelry case

The first two instances would both be connected to the same idea; they both use the iPod as a container (the smallness is implied in the former by the size of the device itself). However, the third and fourth instances both encode additional information, and thus are not considered the same idea as the first two. In total, the example above includes four instances and three ideas. The explicit criteria for separating ideas is described below.

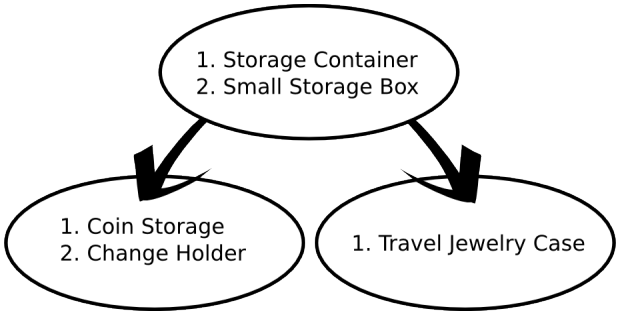


Figure 3.3: Example category tree

3.3.2 Idea forests and deriving quantity

While the above differentiation between instances and ideas is useful for a conceptual derivation of quantity, it does not provide an explicit means for disambiguating ideas. Furthermore, it neglects another source of granularity in brainstorming datasets: ideas share elements between them. For example, consider instances 1 and 3 in the above list of examples. These ideas both use the shell of an iPod as a container, but are not semantically equivalent. The second idea encodes additional information: the use of that container as coin storage. Idea 1 is a *generalization* of idea 2.

Generalization is a common relationship between brainstorming ideas, occurring most often between responses by two brainstormers. It is desirable to encode this relationship between ideas in our representation while maintaining their disambiguation. To do this, I expand the "idea" side of the graph (Figure 3.2) into a tree structure. Instances still link to ideas, but ideas may now also have parent-child relationships. An idea A is a parent of an idea B if all instances connected to A are generalizations of all instances connected to B. Put another way, instances in B encode all the semantic information of instances in A, plus something extra.

An example of this hierarchical relationship between ideas is given in Figure 3.3. Each connected subgraph of ideas is referred to as a *category tree*, or simply a *category*. The example replicates the relationship between solutions to the iPod question using the shell as storage. For a given brainstorming question, there will be many distinct category trees with no common parents. A collection of category trees generated in response to a brainstorming question is known as a *idea forest*.

A collection of idea nodes is a valid idea forest if it adheres to the following four constraints:

1. equivalence: All instances mapped to an idea node should be semantically equivalent.
2. generalization: For each idea A, any instance from any ancestor idea B should be a generalization of any instance from A.
3. common parent: For any two instances from two ideas, if those instances share some semantic meaning, their respective ideas should have a common parent, or one is the parent of the other.
4. non-equivalence: For any two instances from two different ideas, those instances are not semantically equivalent.

Intuitively, the first constraint ensures that each node is a single semantic idea, and correctly disambiguates between instances. Given that the first constraint holds, the other constraints ensure that the hierarchy is valid, there are parents that capture all meaningful relationships, and that exactly one node exists per idea. An idea forest that adheres to each of these constraints can be used to derive quantity of ideas generated simply by counting the number of nodes in the forest.

A summary of the terminology introduced in this section and the previous is given in Table 3.2.

3.3.3 o-score

The o-score metric introduced by Jansson and Smith [29] is a measure for novelty that is quantified based on the number of occurrences of an idea relative to the size of the idea pool overall. This metric captures only novelty, and thus neglects other elements of creativity, such as appropriateness and practicality. However, novelty (alternatively originality) is a component of every creativity measure observed in related work.

The o-score can be deterministically derived given a system for disambiguating ideas. If desired, the resulting novelty metric can be incorporated into a more encompassing measure of creativity after the fact. The determinism of this measure aligns well with the quantitative goals of this research, and as such it is employed throughout. This is not to say that novelty is a sufficient substitute for the entire measure of creativity, but merely that it is the most realistic to derive for large data sets given a means of disambiguation. The problem of integrating novelty into an overall creativity metric is left to future work.

The idea forest provides an explicit means for disambiguation: instances in different nodes correspond to different ideas. As a result, o-score can be deterministically derived according to the definition:

term	definition
instance	A single text response to a brainstorming prompt.
run	A temporal ordering of instances given in response to a brainstorming prompt by a single participant in a single session.
campaign	A temporal ordering of instances given by *multiple workers* to a brainstorming prompt, such that the same number of instances were requested from each worker.
idea	A conceptual collection of instances which represent the same semantic solution to a brainstorming prompt, with rephrasing.
idea mass	The number of instances associated with an idea node.
generalization	An idea is a generalization of another idea if all instances in the former are generalizations of all instances in the latter.
category tree	A subset of ideas which are connected by generalization relationships.
idea forest	A collection of unconnected category trees in response to a single brainstorming prompt.

Table 3.2: Summary of terminology for brainstorming abstractions

$$\text{idea o-score} = 1 - \frac{\text{\# of instances of idea}}{\text{total \# of instances in data set}}$$

Figure 3.4: o-score definition

Throughout this thesis, o-scores will be utilized as a measure for novelty.

3.4 Summary of data collected

The corpus of brainstorming responses to the four questions were coded to produce idea forests. One coder generated idea forests for both the iPod and turk corpora, while each of the other coders completed a single data set (forgot name and charity, respectively). Across all questions, 165 instances (1.8%) were discarded as random, unrelated to the question, or too ill-specified to interpret. The criteria for discarding an instance was that the coder was not able to visualize the solution in the context of the brainstorming problem. This section will provide some brief descriptive statistics of the data sets, as well as visualizations of the idea forest structure. Some characteristics, such as number of trees, are dominated by the number of responses gathered, which varied from question to question. In these (explicitly noted) cases, statistics are normalized by the number of instances collected for the associated question.

To ground the discussion from this point on, I first introduce a visualization for an entire idea forest. Because idea forests are collections of trees, they can be visualized collectively using standard tree-visualization layouts by introducing a root node for which all category trees are children. I chose a circle-packing visualization for this because the spacing allowances make it easy to separate trees visually without a diverse array of colour. Colour-based separation is difficult and loses meaning with the hundreds of idea nodes represented. In the circle-packing visualization, each small, empty circle represents an instance. Circles that contain other circles are thus ideas. The hierarchy of ideas is represented by repeating nested circles. Category trees are circles which are contained in only the outermost circle (the forest). The relative idea mass of category trees can be evaluated by comparing the total number of instance circles. The radius of an idea circle is roughly indicative of the complexity of the category tree, particularly its depth.

Circle-packing visualizations are given for each of the four brainstorming question in

question	number of instances	number of nodes	number of trees
iPod	3005	1220 (0.41)	322 (0.11)
charity	1861	1559 (0.84)	210 (0.11)
forgot_name	2234	601 (0.27)	147 (0.07)
turk	2031	1037 (0.51)	274 (0.13)

Table 3.3: Descriptive statistics for size of idea forests. The brackets indicate the same value normalized by the number of instances collected.

Figures 3.5, 3.6, 3.7, and 3.8. They were generated using the D3.js library[1]. Credit is given to Chris Zheng for the initial implementation of the visualization for this data set.

Descriptive statistics for each of these forests are given in Table 3.3. The same quantity normalized by the number of instances in the data set (i.e. the quantity per instance) is given in brackets, as values such as number of ideas are highly dependent on the amount of data captured.

Also of interest is the depth and breadth of trees in the idea forest, as they can be roughly interpreted as the amount of detail in which ideas are explored, and the amount of variety within a particular category. Depth is obviously quantified as depth per tree, while breadth is calculated as the number of children of non-leaf nodes in category forests. Quartiles for these measures are visualized in Figures 3.9 and 3.10.

3.4.1 Brainstorming runs

The overall form of the idea forest is of interest, but so is the format of individual brainstorming runs. Brainstorming runs can be visualized as ordered lists of nodes in the idea forest. These take the form of a separate set of directed edges that describe a traversal of the idea forest.

One phenomenon that is of interest brainstorming runs is that of *riffing*. When brainstorming, participants are likely to return to previous ideas and manipulate them. This is exactly the behaviour predicted by SIAM [47]. I define an instance in a run as a *riff* if it is a descendant or ancestor of any node for which there is a previous instance in the participant's brainstorming results. Conversely, any instance for which a later riff exists is known as a *source* instance if it itself is not a riff, or a *chain* instance if it is also a riff (an instance can be both a riff and a chain). Table 3.4 describes the prominence of each

[1] http://d3js.org/

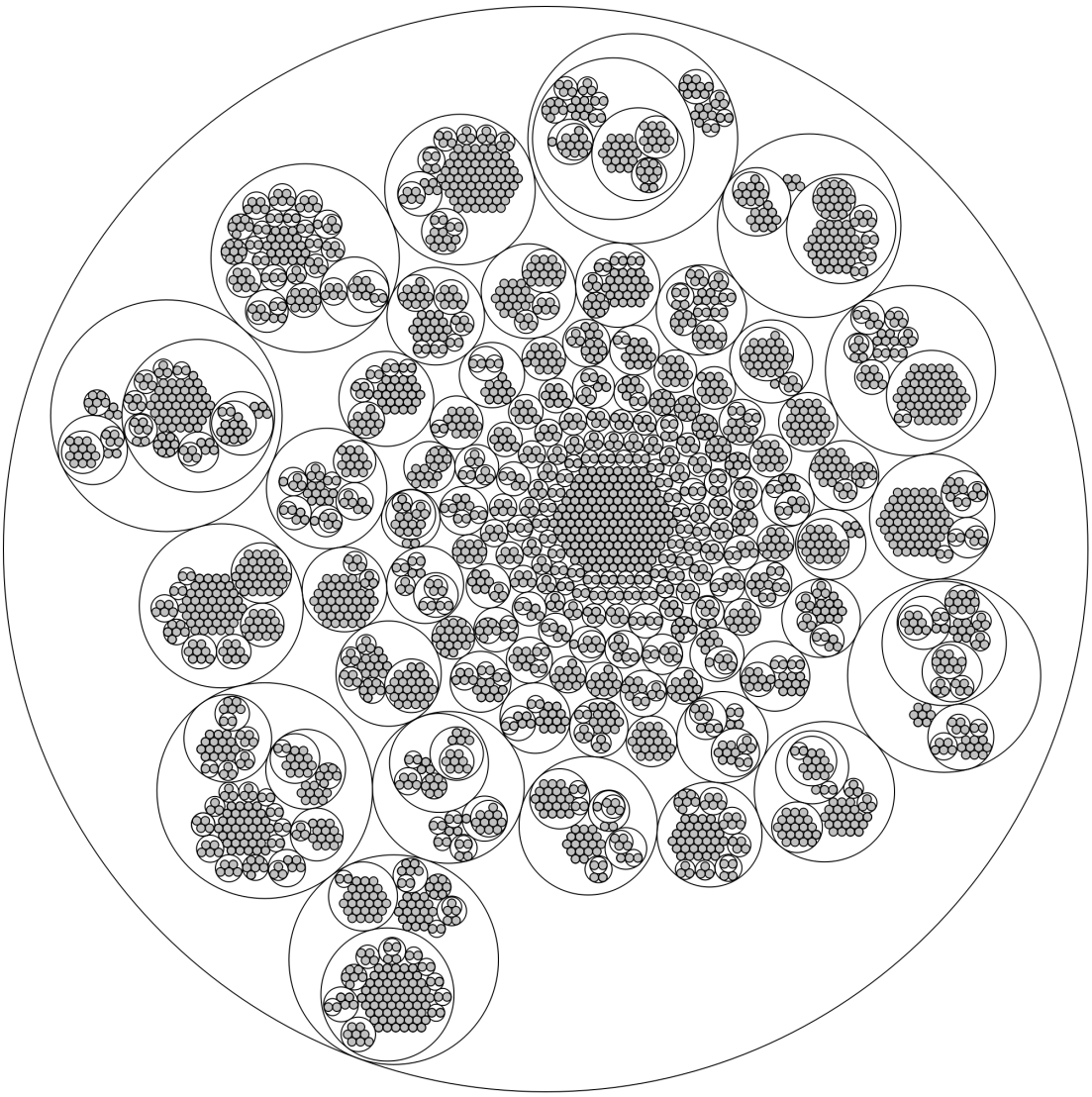


Figure 3.5: Idea forest visualization for the iPod dataset.

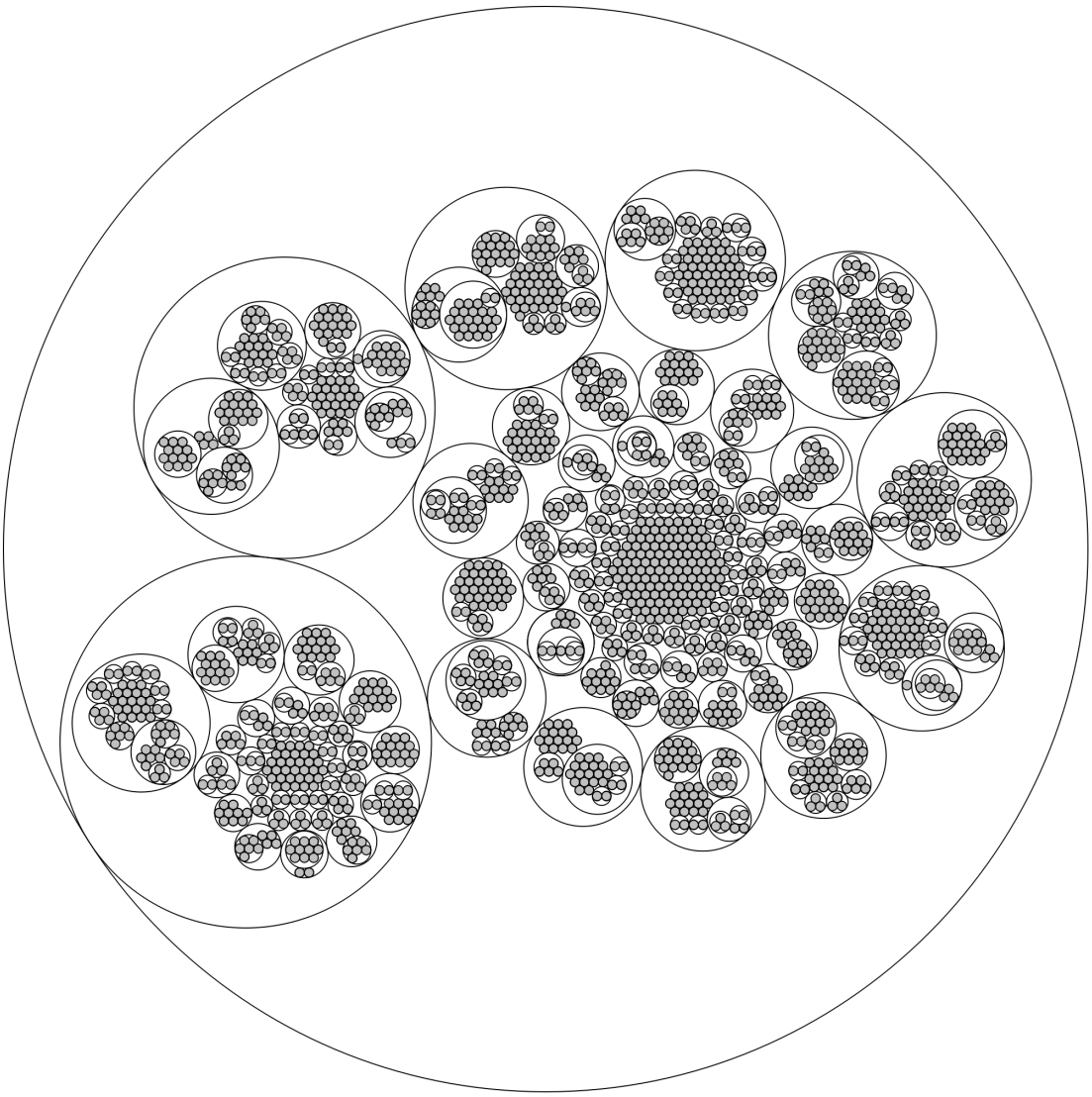


Figure 3.6: Idea forest visualization for the charity dataset.

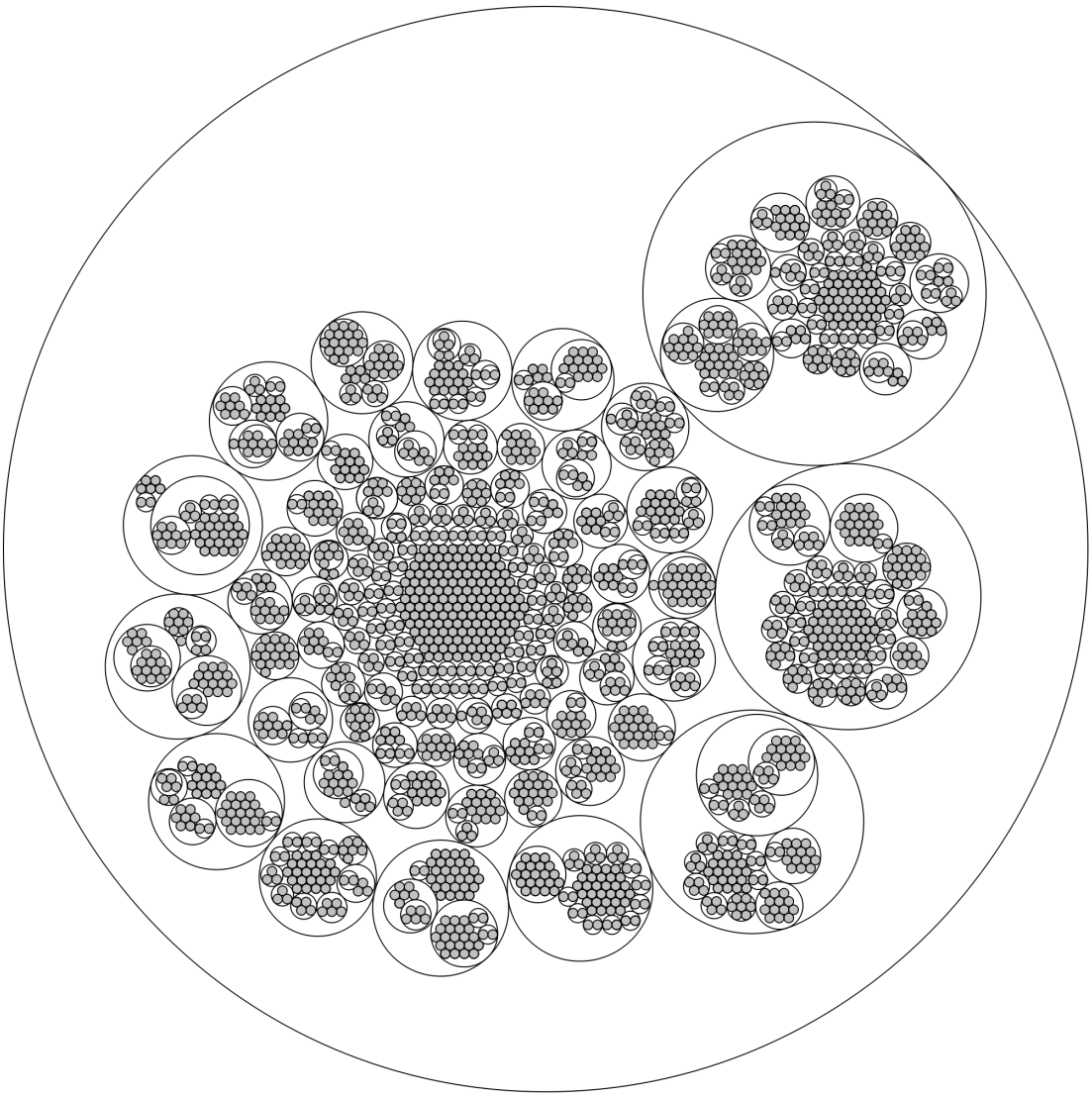


Figure 3.7: Idea forest visualization for the turk dataset.

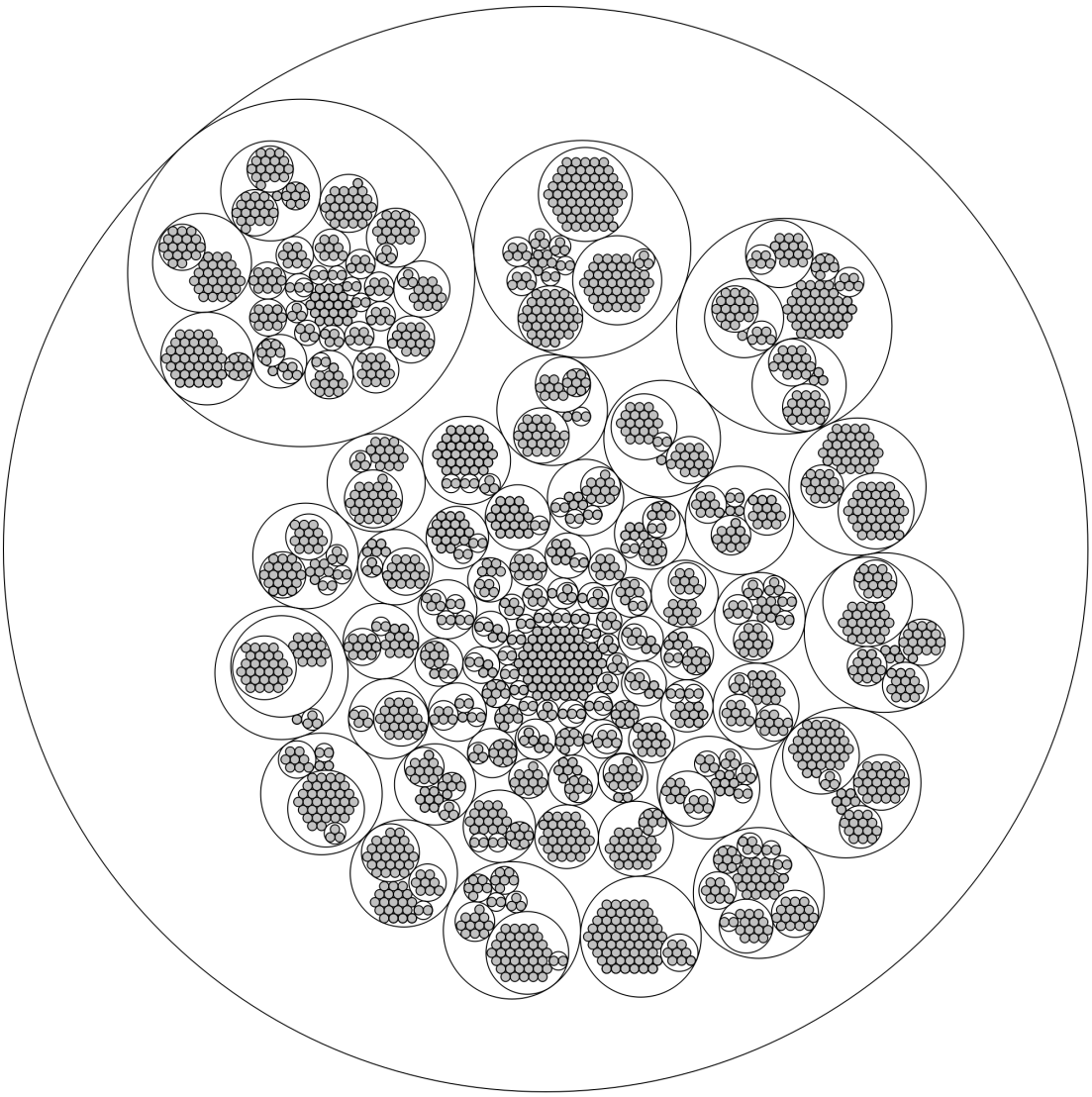


Figure 3.8: Idea forest visualization for the forgot name dataset.

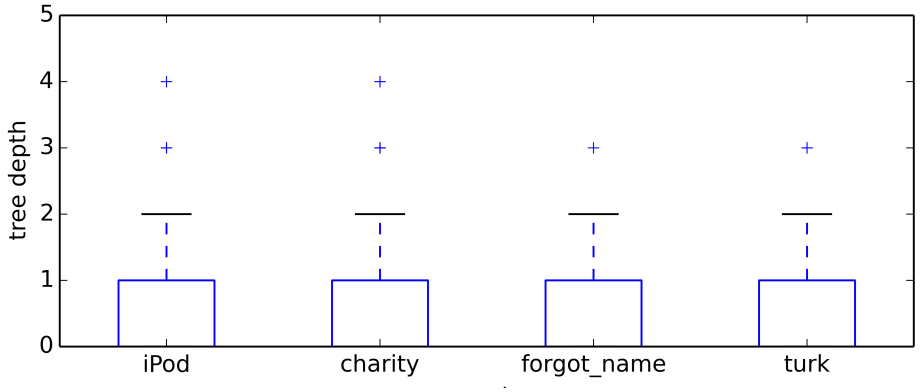


Figure 3.9: Tree depth between questions in the idea forests.

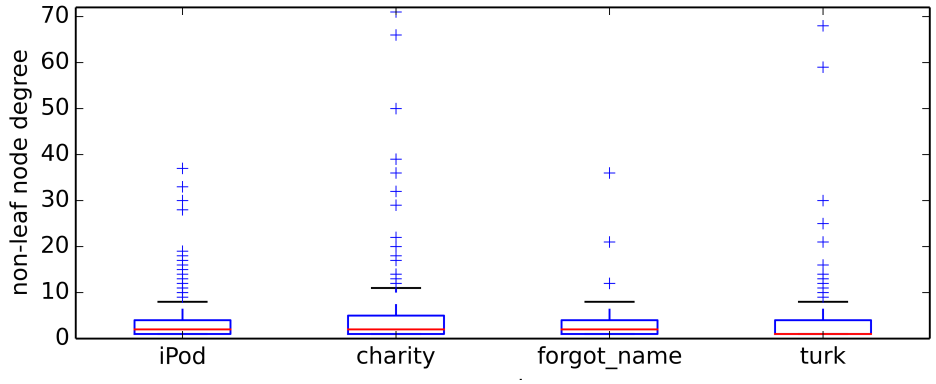


Figure 3.10: Tree breadth between questions in the cluster forests. Calculated only for non-leaf nodes, as the degree of the node minus one.

question	riffs	source	chain
iPod	1457	491	893
charity	1064	289	730
forgot_name	1481	413	1037
turk	1183	310	822

Table 3.4: Run descriptive stats. Each value is the median number of instances with the given characteristic, where counts are normalized by the number of instances given in the run

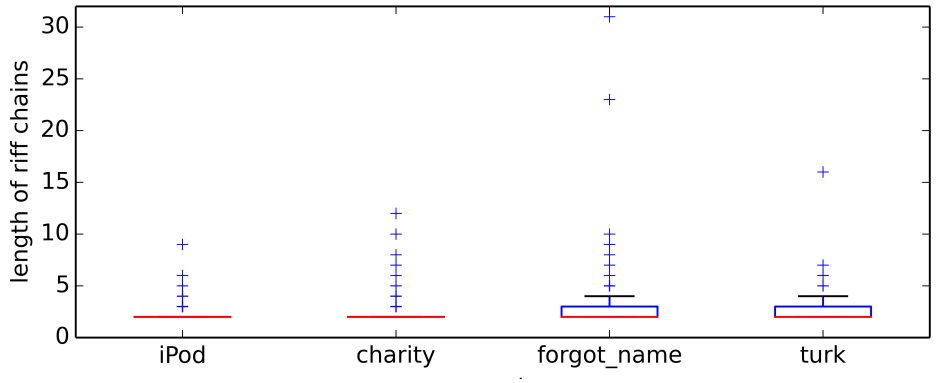


Figure 3.11: Length of riff chains (consecutive instances which share a riffing relationship) for each question.

of these instance types for each question, normalized by the number of instances given. Figure 3.11 demonstrates the interquartile ranges for the length of *riff chains*, consecutive sequences of instances sharing a riffing relationship. Notably, the median length of riff chains was 2 for all problems, suggesting that participants would rarely remain within a concept for more than one idea. Furthermore, the third quartile was at most 3, suggesting that riffing is generally a transient phenomenon.

3.5 Summary

This chapter introduced the study performed to generate a corpus of brainstorming responses. The goals of this corpus were threefold: to provide real data against which models of brainstorming could be tested, to provide a body of text responses from which quali-

tative conclusions could be drawn, and to provide a data set of brainstorming responses with labeled quantity and novelty outcomes that can be used to generate predictive models in future work. The process for collecting this data was described, with a description of how the study design was arrived at. The chapter finished by introducing idea forests as a mechanism to derive quantitative outcome metrics, and giving descriptive statistics for the 10,000-response corpus.

Chapter 4

Construction and validation of idea forests

4.1 Introduction

This chapter describes the methodology employed to construct idea forests and validate the findings derived from them. The contributions of this chapter focus on the benefits of idea forests when applied to brainstorming corpora at massive scales enabled by microtask marketplaces. Thus, it may be safely skipped by readers primarily interested in the theoretical contributions of this thesis. First, an algorithm for the construction of idea forests is given. Second, this chapter introduce a simulation-based approach for validating findings derived from idea forests.

4.2 Generating idea forests for idea disambiguation

The idea forest structure gives a great deal of analytical power for examining a corpus of brainstorming ideas. The intention of this work is to test the most basic assumptions of brainstorming in a microtask environment. One of the unique properties of microtask marketplaces in contrast to traditional brainstorming environments is the scale at which ideas can be gathered. At the scale of thousands of responses, organizing data and ensuring consistency is non-trivial.

I tackled these problems of scale in a variety of methods. First, I applied Natural Language Processing (NLP) and Machine Learning techniques to provide a "first pass"

similarity scoring between instances and clustering of ideas, used to suggest and inform decisions made by coders. Second, I created an algorithmic coding scheme that leverages the hierarchical nature of the idea forest to reduce the scope of judge decision making in the iterative process of idea forest generation.

4.2.1 NLP and clustering

It would be ideal to automatically determine which instances are semantically equivalent. Unfortunately, automatically disambiguating English language phrases is a monumental topic of research unto itself. In knowledge and data engineering, this problem is often referred to as *entity resolution* or *entity linking* [18]. Automated implementations of entity resolution provide insufficient accuracy for application to this problem.

Other attempts at solving this problem have utilized microtask marketplaces in order to leverage human expertise [65, 67, 14]. Wang et al. [65] attempt to tackle the problem of scalability when taking this approach — naively, $O(n^2)$ HITs are required to resolve n entities. Under their technique, comparisons for semantic equivalence are first filtered by a threshold criteria on ML-derived similarity scores before being sent to the crowd. This technique was tested in this thesis, but commonly led to queries of high semantic equivalence being excluded as unrelated or vice versa. Furthermore, given an idea pool of up to 4000 instances per question, even a significant reduction of comparisons led to large (and expensive) HITs that dwarfed the expense of establishing the initial corpus. Furthermore, it is unclear that crowd participants could efficiently be given sufficient training to disambiguate between ideas in a repeatable way.

Thus, I decided that neither automated or crowd-based entity resolution systems could provide a panacea to the problem of disambiguating brainstorming ideas. Instead, many of these techniques are applied to provide suggestions to a coder. In particular, I used NLP techniques similar to those employed by Wang et al's filtering step to provide similarity scores for all $O(n^2)$ idea pairings, and applied a technique called *correlation clustering* [4] to produce a preliminary clustering of instances groups that were potentially semantically equivalent.

Similarity

To compute similarity between instances, each instance undergoes a query-expansion using WordNet [44] via the python NLTK library[1]. Each instance is stripped of stop words

[1] http://www.nltk.org/

and the remaining words are stemmed and placed in a bag of words. Following this, all synonyms and hypernyms are added to the bag of words recursively. I found this approach was more likely to capture semantic relationships between bags of words than simpler metrics such as edit distance [36], simply by casting a wider net to match upon.

This bag of words is then weighted using tf-idf. Because each term is represented only a single time in the bag of words, this amounts to assigning each term i the weight $w_i = \frac{1}{\#\text{ of times word used in all bags}}$. To take the similarity between two weighted bags of words, each is treated as a vector and the cosine similarity is computed. This similarity metric is employed throughout the coding process.

Clustering

The cosine-similarity metric is sufficient for use as a distance score in most clustering algorithms. Unfortunately, many popular clustering techniques assume *a priori* estimates of the number of clusters (for example, k-means clustering). One promising alternative introduced by Bansal et al. [4] is *correlation clustering*. Correlation clustering assumes a complete graph of positive and negative edges, where a positive edge indicates a preference to the same cluster and a negative edge the opposite. The technique optimizes to minimize the number of *disagreements*, counted as pairs of nodes in the same cluster joined by a negative edge or pairs of nodes in different clusters joined by a positive edge.

To apply correlation clustering to the idea resolution problem, each instance is treated as a node, and a positive edge is placed between two nodes if the similarity of their two instances is above some threshold. Through experimentation, I fixed this value to 0.5. For each data set, I computed an automated clustering using a custom implementation of the cautious algorithm described in Bansal et al. [4]. This implementation is available under the package name py_correlation_clustering[2]. While there are certainly more complex implementations of correlation clustering, including those that account for continuous similarity measures and are optimized for large data sets [3], the naive cautious algorithm was sufficient in my experience for producing initial clusters that captured several of the largest idea categories.

4.2.2 An algorithm for idea forest generation

Clustering techniques give a rudimentary approximation of similarity for ideas, as well as a starting point for generating clusters. However, these automated methods are unable to

[2]https://github.com/thefil/py-correlation-clustering

produce complete and correct semantic disambiguation of ideas. To complete the production of an idea forest, a system involving human judges is necessary. To promote consistent judging, an algorithm was produced to transform a series of English language instances into an idea forest. This algorithm iteratively inserts new instances by traversing the existing idea forest, and appeals to a human judge to make decisions directing that traversal. By traversing the tree, only a limited subset of previous instances and ideas need to be considered at each step involving a human judge. There are three algorithm step which require human judgment:

Similarity: In this step, the judge chooses a node which is most semantically similar to the idea being inserted. Formally, given a set of semantically equivalent instances and a set of existing ideas, return the idea containing instances most semantically similar to those in the set. If no ideas contain similar solution elements, return None. The set of ideas presented is limited to those that are children of the current node in the tree traversal.

Generalization: The judge identifies any generalization relationship between two ideas. Formally, given two ideas, one generalizes another if every instance in that idea is a generalization of every instance in the other. Given two such ideas, return whether a) one generalizes the other, b) each generalizes each other, or c) neither generalizes the other.

Artificial node creation: In the case where two ideas are related but no generalization relationship exists, a judge creates a new idea which generalizes both. This can be thought of as creating an idea which contains the intersection of the semantic content of both. Formally, given two ideas, such that neither idea generalizes the other, provide a new idea that generalizes both ideas and is generalized by any parents of these ideas.

In addition, judges are prompted to label idea nodes, which provides does not impact the structure of the forest.

These decisions are employed in a tree traversal algorithm, the gist of which I describe followed by a pseudocode implementation. Traversal begins at the root of the idea forest. Iteratively, sets of semantically equivalent instances are introduced and create a new idea node. The similarity prompt determines if the new instances belong in an existing category tree. If not, a new tree is created. Otherwise, traversal shifts to the root of the selected category tree. Then, this root and the new idea must be compared to determine the local structure of the tree. For this, the generalization prompt is employed. If neither idea generalizes the other, the artificial node creation prompt creates a new parent replacing the root under which both ideas are placed. If both ideas generalize each other, they are merged. Otherwise, the more general idea occupies the position of the root node, and the other idea becomes its child. The algorithm then repeats recursively with the traversal node

acting as the root of an idea forest. The pseudocode algorithm is specified in Figure 4.1.

Proof of algorithm

Assuming an oracle for the judging tasks, it can be proven the result of this algorithm produces a tree structure meeting the four constraints for an idea forest:

1. equivalence: All instances mapped to an idea node should be semantically equivalent.
2. generalization: For each idea A, any instance from any ancestor idea B should be a generalization of any instance from A.
3. common parent: For any two instances from two ideas, if those instances share some semantic meaning, their respective ideas should have a common parent, or one is the parent of the other.
4. non-equivalence: For any two instances from two different ideas, those instances are not semantically equivalent.

I apply proof by induction. In the base case, there is a single node in the idea forest. The instances in this node must be semantically equivalent, as only semantically-equivalent nodes can be added (alternatively, the algorithm can be restricted to adding a single instance at a time). There are no generalization, artificial parent, or non-equivalence relationships possible, so this is a valid idea forest.

For the inductive step, I show that given a valid idea forest, the addition of an additional idea node cannot violate any constraints. Each constraint is first considered for inserting a new regular node, and then for inserting a new artificial node. First, consider adding a new regular node.

For equivalence, I provide a proof by contradiction. Assume a node is added such that the constraint is violated. Then either the new node contained instances that were not semantically equivalent, or two semantically non-equivalent nodes were merged. The first case is again precluded by input constraints. For the merge case, nodes are only merged if they generalize each other. Two semantically different ideas cannot full describe each other, therefore this is impossible. Thus, equivalence must hold.

For generalization, I again provide a proof by contradiction. If the node is merged with an existing node that satisfies constraints, then the contradiction is trivial. Otherwise,

```
1  for each instance:
2    idea_node = new node including instance
3    current_node = root of forest
4    do:
5      best_match = similarity(idea_node, current_node.children)
6
7      if best_match.similarity is None or current_node has no children:
8        insert idea_node under current_node
9        exit do
10     else:
11       if generalization(idea_node, best_match) == both:
12         merge idea_node, best_match
13         exit do
14       else if generalization(idea_node, best_match) == neither:
15         new_parent = new artificial idea node
16         insert best_match, idea_node under new_parent
17         insert new_parent under current_node
18         exit do
19       else if generalization(idea_node, best_match) == idea_node more
             general:
20         replace best_match with idea_node in tree
21         current_node = idea_node
22         idea_node = best_match
23       else if generalization(idea_node, best_match) == current_node
             more general:
24         current_node = best_match
```

Figure 4.1: Idea forest generation algorithm

three cases are possible: the new node is not generalized by an ancestor, the new node does not generalize a descendant, or the new node is not generalized by a new artificial node. In the case where the new node is not generalized by an ancestor, the only way for the new node to be inserted below the ancestor would be for that ancestor to be identified as a generalization (the fourth case of the generalization conditional), which is a contradiction. In the case where the new node does not generalize a descendant, the new node must have been inserted as a parent of an existing node which is a generalization of the descendant. This can only happen in the third case of the generalization conditional, which requires that the new node is a generalization of this existing node, satisfying the requirement and creating a contradiction. Finally, in the case where the new node is not generalized by a new artificial node, the new artificial node must generalize the new node by definition when created.

For the common parent constraint, I again provide a proof by contradiction. Assume a new idea is added that shares semantic similarity with another idea but does not have a common ancestor. Then the new idea must have been added as a new category tree. This can only occur if the oracle determines that there is no semantic similarity, which is a contradiction.

For non-equivalence, I again provide a proof by contradiction. Assume that the new node was not merged with a semantically equivalent node. These nodes must generalize each other, so if they were ever compared, they would be merged. They must be compared, since the new node must take the same path through the idea forest that the original node took, since all decisions are based on semantic meaning, which is identical. Therefore, they would be merged, and there is a contradiction.

Second, consider adding an artificial node.

For equivalence, the new artificial node is empty of instances, so it by definition has no un-equivalent instances.

For generalization, by definition artificial nodes are inserted to generalize their children and be generalized by their parents.

For common parent, since the new artificial node has all the same ancestors as its children, and has only the intersection of the semantic meaning of its children, this constraint must hold.

For non-equivalence, adding an artificial node does not introduce any new instances, so it cannot be equivalent to any other ideas.

Thus, adding a new node to a valid idea forest ensures that the new forest also satisfies the constraints, and the idea forest is valid by induction.

■

Of course, human judges do not operate as oracles. They make mistakes and have disagreements as to the semantic relationships between instances. The validity of idea forests resulting from employing human judges is explored in Section 4.3. In the next section, I will discuss integrating this algorithm and the NLP and ML methods described above into a set of tools to support judging.

4.2.3 Tool support

This algorithm for idea forest generation is integrated into a wizard application.A screenshot of the application is given in Figure 4.2.

In the wizard application, the left panel provides a list of instances that have not yet been inserted into the forest, while the right provides a collapsible view of the current idea forest structure. Nodes in the forest are represented by a judge-supplied label if available, otherwise the text of the first instance associated with that node.

Judges can select multiple instances in the left panel (they are required to adhere to the input standard of semantic equivalence) and initiate the algorithm described in the previous section. The system automates the insertion, prompting the human coder asneeded for the judging tasks. In the case of the similarity prompt, a list the five existing idea nodes with the highest automatically-generated similarity score to the new node are provided as suggestions.

To facilitate this, they may sort the instances according to the cosine similarity metric. By default, the instances are sorted according to the clustering provided by the correlation clustering algorithm, with larger clusters near the top. No explicit delineation between clusters is given. In practice, this clustering and sorting functionality dramatically increased the rate of coding for the first third to half of a brainstorming corpus, after which the similarity function could no longer identify semantically meaningful relationships between instances.

The application also allows for some transformation operations on the forest to allow judges to correct any errors made in the judging steps of the algorithm. These operations include introducing new artificial nodes, re-parenting, and removing nodes or instances from the forest. These were introduced after judges described frustration with having to re-execute the clustering algorithm when their judgment regarding a single semantic relationship had changed. For example, to return to the example at the beginning of this

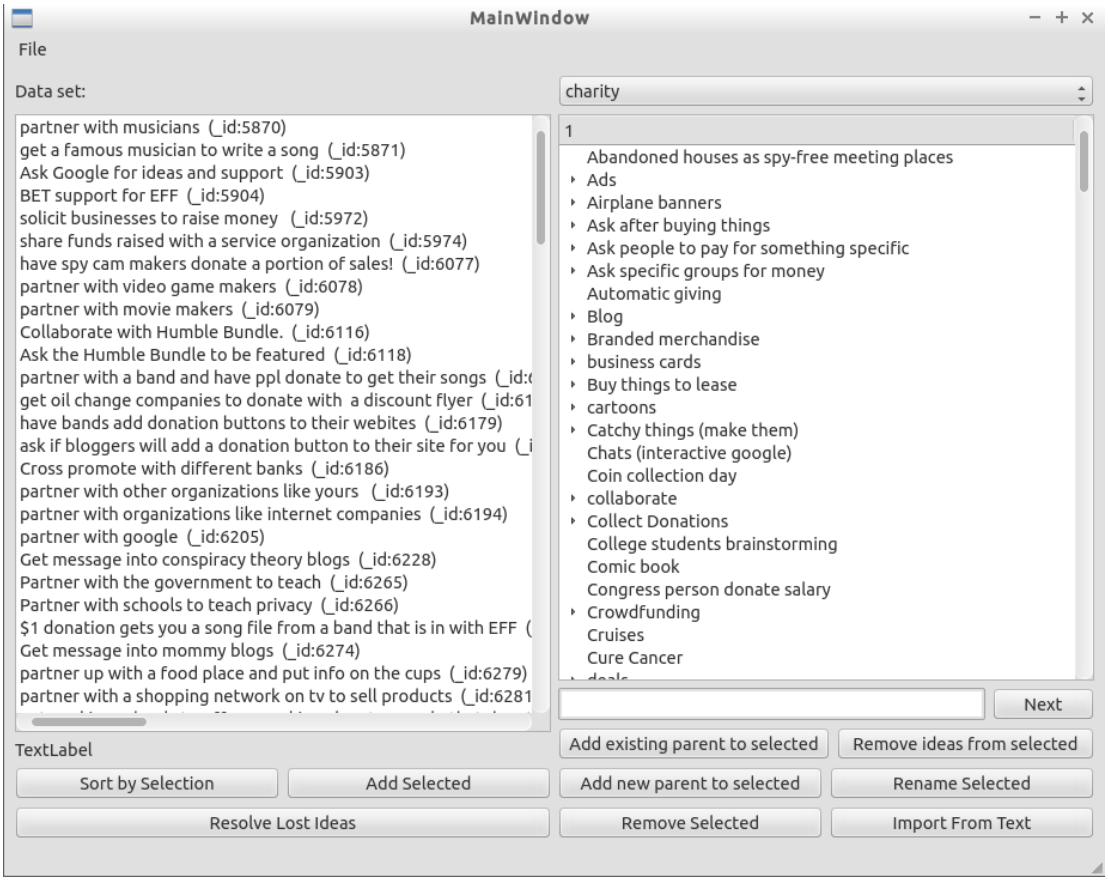


Figure 4.2: Clustering wizard

section, a judge may determine that the "travel" element of "travel jewelry case" represents a stronger semantic link than the jewelry aspect, and move the corresponding node from under "jewelry" to under "travel case". These multiple-generalization relationships were not exceedingly common, but do suggest that it may be appropriate in future work to consider expanding the idea forest concept to allow multiple parents.

4.3 Validity of idea forests

The cluster forests are large data structures, and determining validity poses a similarly oversized challenge. Having multiple judges code the entire data set requires extensive amount of time, and even were two judges to produce a coding, there is no pre-existing inter-rater reliability measure for a data structure as described. This section describes a novel simulation-based method used to verify the semantic validity of the data structure. First, I adopted a random-sampling approach to determine error rates for each constraint that the forest should satisfy. Following this, I simulated the impact of these errors on the data set when conducting analysis.

4.3.1 Estimating error rates

Disambiguating creative ideas is a difficult task that results in some disagreement between disparate judges. That said, there are four constraints which should hold in a complete idea forest. These constraints are reproduced from Section 3.3.2:

1. equivalence: All instances mapped to an idea node should be semantically equivalent.
2. generalization: For each idea, any instance from any ancestor idea should be a generalization of an instance from that idea.
3. common parent: For two instances from two ideas, if those instances share a common property, those instances should have a common parent, or one is the parent of the other.
4. non-equivalence: For two instances from two different nodes, those instances should not be semantically equivalent.

To quantify the validity of the data set, I first had to estimate the probability of any of the constraints being violated. This can be decomposed into two parts. Constraint 1 is considered only in the context of a single node. Constraints 2-4 assume that constraint 1 holds, and then evaluate relationships between nodes rather than single nodes.

To estimate the probabilities of these errors, I generated random pairings of idea nodes from each idea forest. A subset of instances for each node selected are evaluated for violating of constraint 1. Then, the pairings themselves are evaluated for each of constraints 2-4. This evaluation takes the form of a judging task in which judges are recruited who did not participate in the coding process. Each judge is considered to be oracular, so the set of violations they identify can be treated as an upper bound on the error rate in the data set.

To generate the set of pairings to be examined by the judges, nodes are binned based on the idea mass of that node and all its descendants. Because most idea nodes are associated with very few instances, it is likely that a completely random sampling of idea nodes would be composed primarily of these small nodes. However, it is expected that nodes with high idea mass would have the most influence on any statistics or modeling. To compensate for this, nodes are binned based on idea mass, and 10 pairings are sampled from each pairing of bins. I fixed the number of bins at 5, both because larger counts resulted in sparse bins, and for the logistical consideration that the number of judging tasks per judge increased as a function of the number of bins. A visualization of this binning of node pairings for one idea forest is given in Figure 4.3.

As demonstrated in the figure, all pairings of all nodes are assigned into bins based on the idea mass of each of those nodes. However, when sampling pairings, only half the space is sampled, as it is symmetrical. 10 pairings are sampled from each of these bins. From these, the judging task is assembled. Each judge is given a subset of the sampled pairings and asked to choose from multiple-choice responses which can be interpreted to identify violations. The instructions given to the judges follow:

> *For each of the following questions, you will be presented two groups of three ideas each. Each idea was given in response to this brainstorming task:*
>
> *(question text)*
>
> *First, for each group, put a small X beside any idea that is not the same idea as the others (with allowances for rephrasing). If none of the ideas are the same, mark them all with Xs.*
>
> *Then, mark one of the 5 options for relationships between group 1 and group 2 of ideas.*

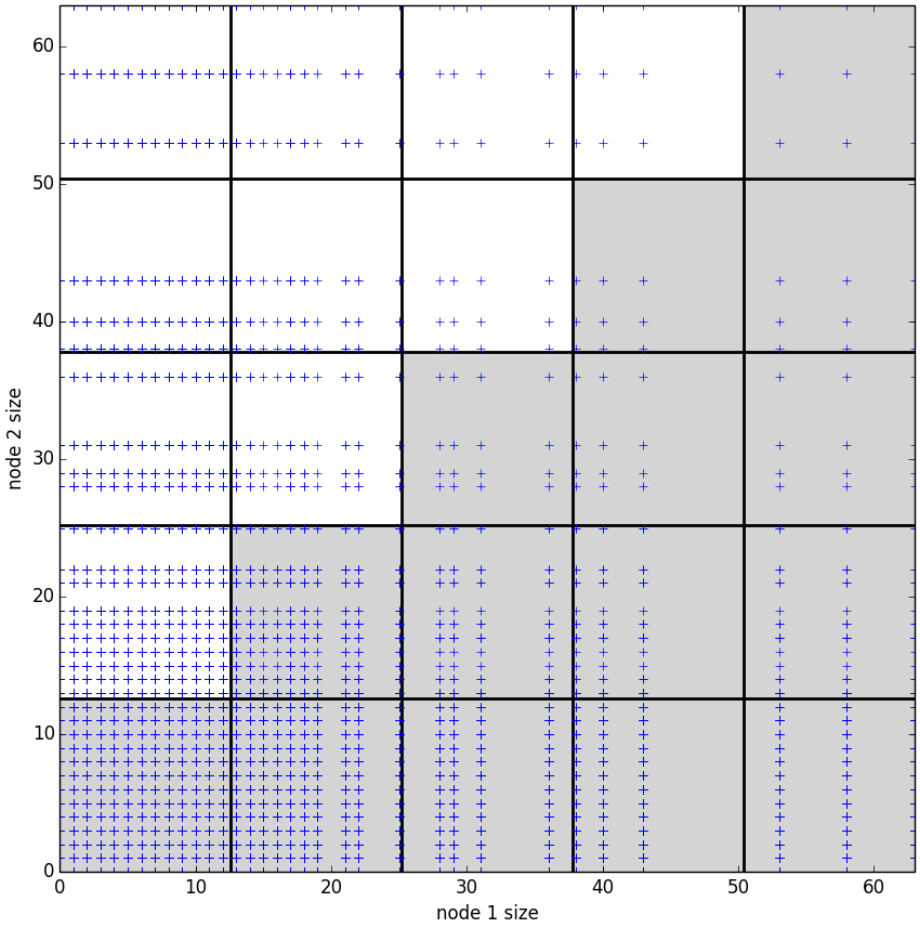


Figure 4.3: Binning idea node pairings based on the idea mass of the node and its descendants. Equal samples are taken from each square in the shaded space, as it is symmetrical. The example given is for the iPod dataset.

Following the instructions, the judges received a series of node pairings printed in three parts. The first and second parts listed a random sampling of instances from the first and second node respectively. The third part gives a list of possible relationships between the nodes. For example, a task for the turk question:

Group 1:
[ ] alerts when new HITs available
[ ] text when new hit available
[ ] Being able to install apps that alert when certain HITs come available

Group 2:
[ ] Updates for mobile app should not be mandatory but recommended
[ ] ability to turn off auto update

Relationship between groups:
[ ] group 1 ideas are generalizations of group 2 ideas
[ ] group 2 ideas are generalizations of group 1 ideas
[ ] group 1 and group 2 are unrelated
[ ] group 1 and 2 are related, but not the same, and neither is a generalization of the other
[ ] group 1 and group 2 are the same

Each judge for the same question was also given a set of 10 common pairs to compare, as a means of computing inter-rater reliability.

For the first two questions, a constraint violation is recorded for each checked box. For the relationship question, constraint violations occur if the relationship identified by the judge and the relationship in the idea forest are non-equal. Rules for determining if a constraint is violated are summarized in Table 4.1. Note that the third checkbox in the relationship question does not explicitly reference a constraint; it is merely the negative case of the generalization and common parent constraints.

From these violation judgments, the error rate for each constraint can be estimated. Each combination of constraint and bin pairing is modeled as a Bernoulli random variable with an uninformed beta prior. In the first two questions (Group 1 and Group 2), each checkbox is a Bernoulli trial for the equivalence constraint and the associated bin. In the latter question (Relationship between groups) each is a Bernoulli trial for each of constraints

constraint	violation conditions
generalization	The judge selects that either group is a generalization of the other, or the idea forest implies one is an ancestor of the other.
common parent	The judge selects that the nodes should have a common parent, or the nodes have a common parent in the idea forest.
non-equivalence	The judge selects that the nodes are identical

Table 4.1: Violation conditions for constraints

constraint	**judges**	**pairs**	equivalence	generalization
turk	5	37	0.12 (0.10, 0.14)	0.06 (0.02, 0.09)
forgot name	3	49	0.22 (0.19, 0.26)	0.02 (0.00, 0.05)
iPod	5	40	0.04 (0.02, 0.05)	0.03 (0.01, 0.06)
charity	3	50	0.06 (0.04, 0.07)	0.03 (0.01, 0.06)
constraint	**judges**	**pairs**	common parent	non-equivalence
turk	5	37	0.06 (0.03, 0.10)	0.01 (0.00, 0.03)
forgot name	3	49	0.14 (0.08, 0.20)	0.02 (0.00, 0.04)
iPod	5	40	0.08 (0.04, 0.13)	0.02 (0.00, 0.04)
charity	3	50	0.13 (0.07, 0.19)	0.01 (0.00, 0.02)

Table 4.2: Idea forest error rates

2-4 in the bin pairing associated with the node masses. A violation of the constraint is considered a success.

A separate survey was conducted for each question. Judges were recruited from the University of Waterloo graduate student population. The number of judges per condition, pairs examined per judge, and resulting error rates (over all bins) are given in Table 4.2. HDIs for the error rates are given in brackets. Note that although the error rate is reported over all bins, per-bin error rates are used in error simulations. They are omitted in the interest of brevity. These violations assume the correctness of the judges in the case of disputes, thus the resulting error rates may be higher than what is actually represented in the data set. This over-estimation is exacerbated by the user of uninformed priors for underpopulated bins. For example, the largest bin pairing in the iPod question has only 3 node pairings for which no judges recorded an error, but this data is insufficient to overcome the prior.

Following the judgment task, the posterior Bernoulli parameters are taken as estimates for:

$$P(E_i|b_1, b_2)$$

Where

$$E_i = \text{A random variable taking the value 1 if constraint i is violated} \tag{4.1}$$
$$b_1 = \text{the bin of the first node in a pair} \tag{4.2}$$
$$b_2 = \text{the bin of the second node in a pair} \tag{4.3}$$

4.3.2 Simulating error impact

The purpose of establishing error rates for constraints is to estimate the impact of errors in the idea forests on the quantitative analysis in the following chapter. When performing quantitative modeling of the corpus, I repeatedly generate manipulations of the idea forests by simulating the changes necessary to fix errors in accordance with the discovered error rates.

For models and hypotheses (which will be identified as they are presented below), statistical tests are run 11 times — once with the cluster forest as produced by the coder, and ten times using permutations of the cluster forest produced by simulating *corrections to errors* according to these empirically derived error rates. Then, the analysis comments on the number of times the same quantitative finding occurs in the forest permutation as in the original forest.

The full algorithm for tree permutation is given in Figure 4.4, but it is briefly discussed here. First, for each node in the tree, each instance in that node is inserted into a new idea node with probability $P(E_1|b_1)$.

After this, for each node in the tree, the probability of error for each of constraints 2-4 given the node bin is estimated ($P(E_i|b_1)$). This is computed by marginalizing $P(E_i|b_1, b_2)$. The error rates for node pairings must be marginalized in this way to prevent an exponential counting of errors; if changes were simulated for each node pairing instead of each node, each node's position would be "corrected" by the simulation up to n times.

A random value is sampled to determine if an error occurs with probability $P(E_i|b_1)$. If so, an error type for the node is randomly sampled from the marginalized distributions,

with the probability of no error being $1 - \sum_i P(E_i|b_1)$. The cluster forest is manipulated to simulate a correction of the error. For example, if the sampled error was of constraint 2 (a parent node was not a generalization of its child), the child node would be removed and placed into its own new tree.

The resulting forest is of the same form as the original, but has undergone transformations similar to those that would be required to fix errors with prominence described by the computed error rates. An example of how idea forests are transformed by the random error simulation is given in Figure 4.5. In this figure, the original iPod cluster forest is shown, along with three permutations of the forest resulting from the error simulation algorithm. It can be seen that the permuted forests contain new, sparse but wide trees, a result of the relatively high common parent error rate. The algorithm results in many new parental relationships being introduced, connecting small trees into sparse trees with greater depth.

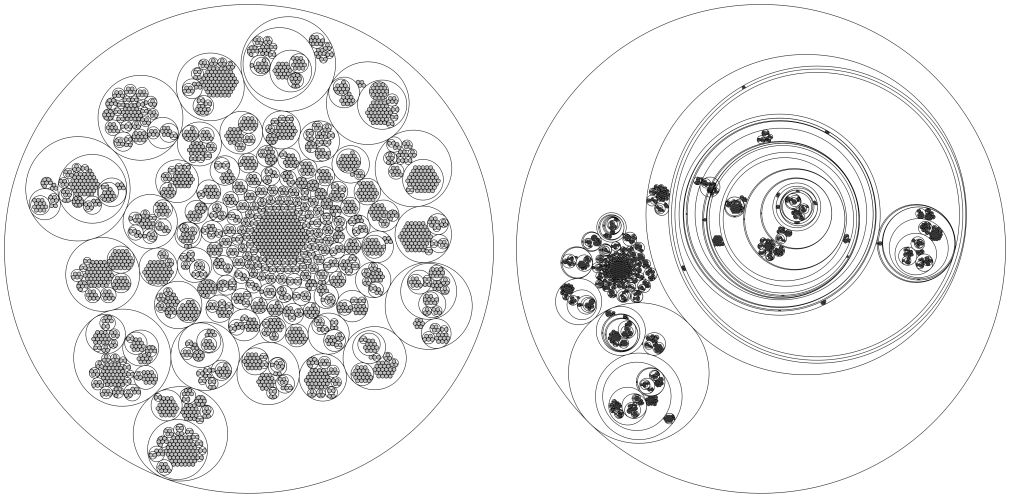
Throughout the remainder of this work, I will refer to the results of models and hypothesis tests both in terms of primary results and results under error simulation. The latter refers to the number of times the model parameter or hypothesis test retained the same outcome under the cluster forest simulated error permutations described in this section. In the case where model parameters are examined rather than the performing of a specific hypothesis test, it is verified that the mean posterior parameter of interest remains within the credibly interval of the posterior parameter distribution in a tree with simulated error.

4.4 Summary

This section has described the methodological approach for data capture and analysis in this work. I provided a method and tools for the construction of idea forests and defined a methodology for establishing the validity of associated statistical findings.

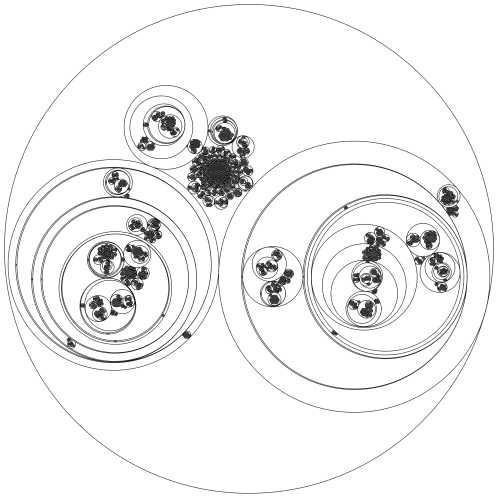
```
1
2  for each node (n1) in forest:
3      for each instance i in n1:
4          x = random float from 0 to 1
5          if x < P(E_1 | |n1|):
6              insert i in new node n2
7              insert n2 at root of idea forest
8
9  for each node (n1) in forest:
10     if n1 has already been deleted:
11         continue
12
13     x = random float from 0 to 1
14     for i = 2 to 4:
15         p_error_i = P(E_i | |n_1|)
16         if x < p_error_i:
17             n2 = get_error_n2(n1)
18             introduce_error(n1, n2, i)
19         x -= p_error_i
20
21 function get_error_n2(n1, i):
22     x = random float from 0 to sum_over_j(P(E_i | |n_1|, j))
23     for j = 1 to total number of bins:
24         p_error_i_j = P(E_i | n1.bin, j)
25         if x < p_error_i_j:
26             return random node sampled from bin j
27         x -= p_error_i_j
28
29 function introduce_error(n1, n2, i):
30     if i == 2: // generalization constraint
31         if n1 and n2 are connected:
32             choose random node n3 along chain from n1 to n2 (excluding the highest
                   ancestor)
33             remove n3 from its parent
34         else:
35             randomly parent n1 to n2, or vice-versa
36     else if i == 3: // common parent constraint
37         if n1 and n2 share an ancestor:
38             choose random node n from n1, n2
39             choose random node n3 along chain from n to that ancestor
40             remove n3 from its parent
41         else:
42             get root of n1 and n2, n1r, n2r
43             create new node n3
44             parent n3 to n1r, n2r
45     else if i == 4: // non-equivalence constraint
46         merge n1 and n2, randomly selecting which parent to keep
```

Figure 4.4: Error simulation algorithm. $|n|$ refers to the bin for node n.

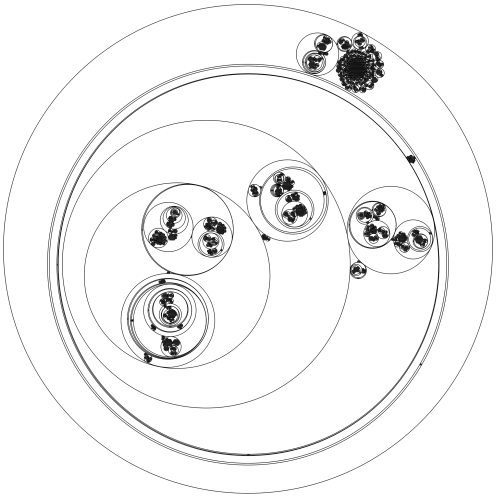


(a) original

(b) permutation



(c) permutation



(d) permutation

Figure 4.5: Example permutations of an idea forest. The upper-left forest is the original data set. The remaining three forests show the forest after it has undergone a randomized simulation of error corrections.

Chapter 5

The quantitative modeling of brainstorming

5.1 Introduction

Two of the primary outcomes of interest for brainstorming are quantity and novelty. Based on the idea forest representation described in the previous chapters, quantity is measured as the discrete number of ideas produced as disambiguated by idea nodes, while novelty is measured as the o-score of an idea node. These measurements enable this chapter, the body of which makes up the major contributions of this thesis.

The goal of this thesis has been to establish baseline models of brainstorming activity in microtask marketplaces. In this chapter, models for the brainstorming outcomes of quantity and novelty will be introduced. Models in this context serve multiple purposes. In this thesis, the posterior parameters of the models on the collected corpus will be examined to make inferences as to the properties of brainstorming in microtask marketplaces. These properties directly address open questions in crowd brainstorming task design.

To extract this information, however, models first had to be derived such that they described both the process of brainstorming and the outcomes accurately. Thus, the models in this chapter should not be taken as one-off constructions to derive a single finding, but rather contributions in themselves which can now be applied in future brainstorming research as a mechanism for testing for statistical differences in outcomes.

I will first briefly describe notation and guidelines for the models following. Then, the models will each be motivated and described in detail. Statistical tests of the poste-

rior parameter distributions of these models will be used to understand the properties of brainstorming in this medium. Specifically, these models will examine the rate at which new ideas are generated, the impact of individuals on the brainstorming process, and how novelty varies as a function of the number of responses given by a participant. I find that the rate of new idea generation decays, that differences in participant ability can result in dozens fewer or greater ideas, and that participants generate their most novel ideas after 18 responses. Finally, the chapter will close with a replication of previous brainstorming findings and a brief survey of the differences in quantitative outcomes observed between brainstorming prompts.

5.2 Modeling practices

The models and statistics presented in this chapter utilize techniques of Bayesian data analysis [34]. Model specifications are given as though they describe a deterministic, generative process for which measurements encode random noise. To describe these models, I adopt the conventions used by sampling software packages such as Stan [61]. The outcome variables of the models are conceptualized as random distributions, such that $y \sim \mathcal{N}(\mu, \sigma)$ should be read as "the outcome variable y is generated by a random process which is described as a normal distribution parameterized by μ and σ". Once this process is envisioned and specified in this notation, models parameters are *fit* to data extracted from the idea forest corpora using Markov Chain Monte Carlo (MCMC) traversal of the parameter space. The distribution of this parameter space traversal estimates the distribution of the posterior of Bayes Rule, where the likelihood is specified according to the model definition and a prior distribution is specified over the parameter space.

In this chapter, these parameter posteriors are used to make inferences as to the nature of brainstorming in microtask marketplaces. I represent the posterior parameter distributions in terms of mean and highest density interval (HDI), where the HDI is computed as all values of the distribution which exceed such minimum frequency, with that minimum chosen such that the integral over those values exceeds 95%. Models are specified and fit using the Stan MCMC language and sampler [61]. A concern when using walks of the parameter space to fit models is convergence. To evaluate convergence, models are fit in 3 chains (independent walks), and the posterior parameter distributions of each chain are visually validated by their similarity. All the models presented in this chapter meet this basic criteria for convergence. In general, the model statements in this chapter are limited to likelihood functions and occasionally the most relevant priors. However, a full specification of each model including parameter bounds and full priors is given in Appendix A.

When models are compared, they are combined into a mixture model with a weighting parameter specifying how much each model contributes to the likelihood function. The posterior distribution of the weighting parameter is examined in consideration of the applicability of the respective models. It should be noted that this approach, though common, is also controversial in the Bayesian data analysis community for relying on explicit model comparison as opposed to model expansion [24].

5.3 Quantity models and rate of idea generation

A primary motivation established in the introduction was to design tasks which optimize the rate of idea generation. It is thus necessary to quantify the number of ideas or categories generated as a function of progress through a brainstorming campaign. Quantity has been shown to correlate highly with quality in prior research [9, 52, 51, 59]. Bouchard and Hare found that the number of ideas generated as group members increased grew linearly [8], but this is unintuitive as there is an expectation of overlap between brainstormers, and thus for the rate of idea generation to decay. Modeling the rate of idea generation allows this intuition to be explicitly tested. It also allows the quantification of the differences in worker productivity; if some workers are more productive than others, then perhaps tasks can be designed to improve lower bounds of productivity or filter workers in some way. Furthermore, if the rate at which the quantity of unique ideas increases can be modeled, then the impact of later interventions can be quantified in terms of the parameters of these models.

Given the idea forest data structure established in the previous section, it is fairly straightforward to quantify idea rate. The number of ideas generated at any point in the brainstorming process is the number of unique idea nodes of the idea forest for which an instance has been acquired. The number of categories generated is the number of unique category trees for which an instance has been acquired. In this section, these values are considered in the context of an entire brainstorming *campaign* on a microtask marketplace. Each condition in the data gathered represents a distinct campaign in which each participant's responses are ordered based on time of submission. An example of campaigns conditioned on question asked is given in Figure 5.1. In this section, the models introduced aim to accurately represent these campaign lines.

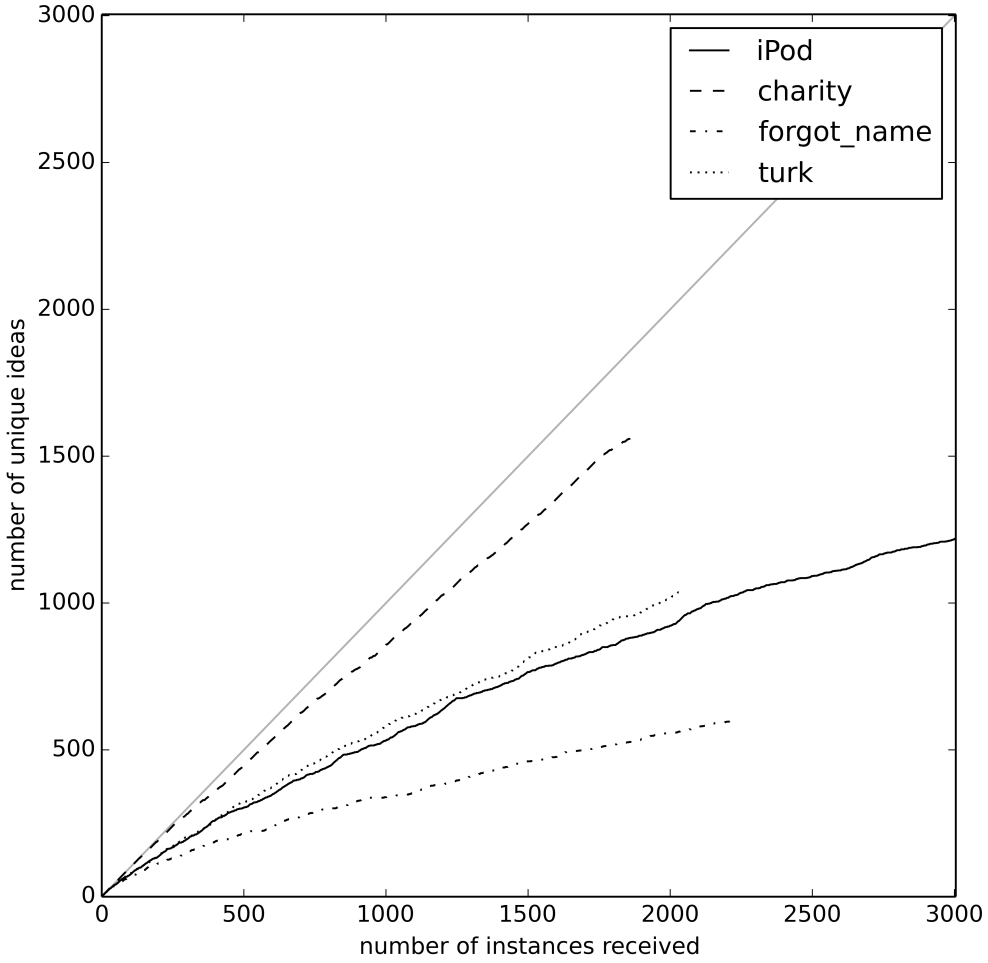


Figure 5.1: The number of unique ideas as a function of number of instances collected, for each question dataset.

5.3.1 Exponential model

It is natural to expect the rate of new ideas to fall over time, and indeed this is reflected in Figure 5.1. Initially, very few ideas have been collected and every instance received from participants is very likely to be novel. A participant contributing later is more likely to propose ideas for which instances have already been received, as the set of ideas received continues growing.

To test the expectation that the rate of new ideas is dropping over time, a model that can encode both linear and exponential growth is introduced:

$$n = \text{scale} * x^{\text{rate}}$$

Where n is the number of *unique* ideas generated, and x is the number of instances gathered thus far. *scale* is a linear scaling parameter, and allows the model to encode linear growth of ideas should that best describe the data. *rate* is the parameter of primary interest. If it is 1, then participants generate a number of ideas linear in the number of instances gathered, suggesting minimal overlap in ideas between participants, a rejection of our expectation. If it is less than 1, the rate of idea generation falls off over time. It cannot be greater than 1, as it is impossible for participants to generate more than one idea per instance.

The derivative of this model describes the *rate* at which novel ideas are generated as a function of the number of instances. Note that because *rate* can be at most 1, the exponent in the derivative will be 0 or negative, resulting in either constant production of ideas, or decay that increases exponentially in x.

To account for fluctuations in the number of ideas gathered as a result of variation in participant abilities, variation in performance between questions, and noise, the model is extended to sample from a normal variable centered on the true value with an additional variance parameter σ:

$$n \sim \mathcal{N}(\text{scale} * x^{\text{rate}}, \sigma)$$

Future models will skip the step of providing a deterministic model and directly introduce a probabilistic model instead. Uniform priors are specified over bounded ranges for both *scale* and *rate*. The ranges were chosen to encompass all values which were conceivable outcomes of the problem domain. For explicit prior distributions, the Stan language model specification is given in Appendix A.1.

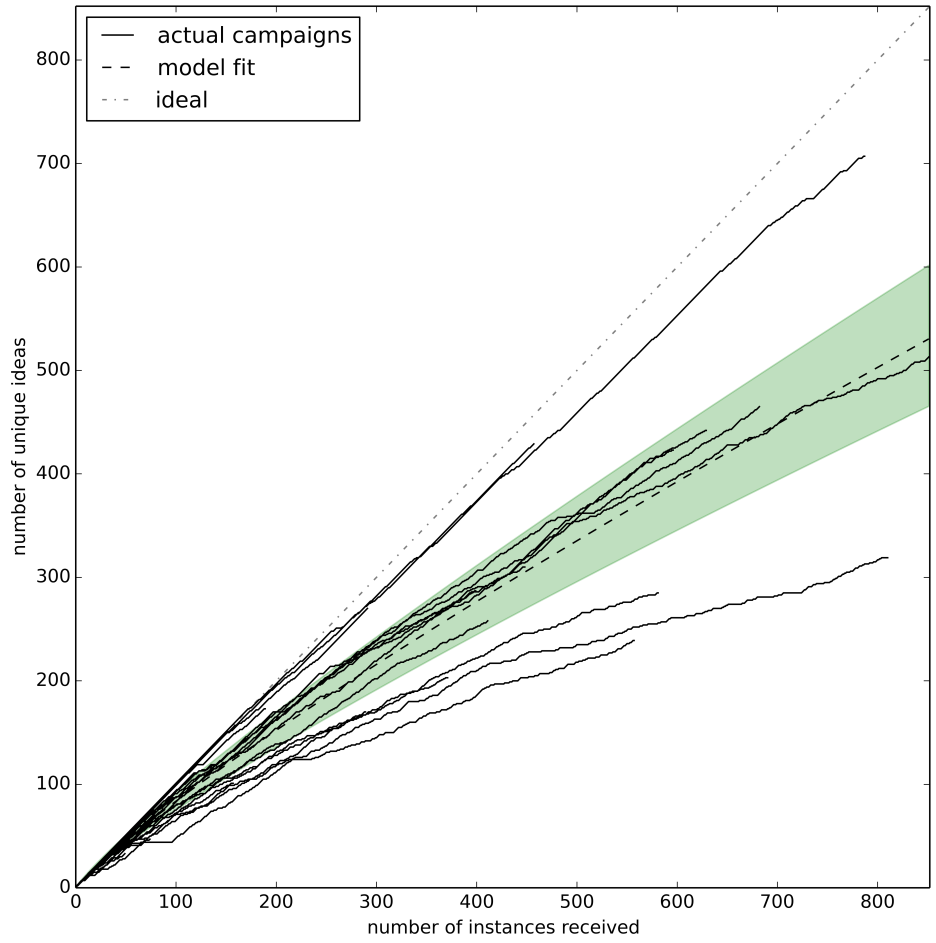


Figure 5.2: Exponential decay model fit: $n \sim \mathcal{N}(\text{scale} * x^{\text{rate}}, \sigma)$
The solid lines represent the actual number of unique ideas as instances were gathered, for each combination of question asked and number of instances requested. The dotted line is the fit model, while the shaded region demonstrates the 95% credible interval of the fit.

The model was fit to the the data for each campaign, where each question asked and number of responses requested condition corresponded to a distinct campaign. The result is given in Figure 5.2. The model converged in all of 3 chains after 1500 iterations. The real data campaigns, which were used to fit the model, are shown as solid black lines. These values were generated for each campaign by ordering all instances first by HIT submission time and then by order in the brainstorming run, and then taking the cumulative *idea* count at each point. The dotted line represents the fit model, the shaded region around which is the 95% credible interval for the fit. Finally, a line representing linear growth is given along the diagonal.

The fit line and HDI represent the posterior belief in the *true* underlying rate of idea generation. Many of the campaign lines fall outside of this. These are accounted for by random noise generated by the normal distribution. The posterior mean of the σ variable is 55.15 (HDI 54.30-55.90), which demonstrates the considerable impact of the noise. This is to be expected, as the model aims to demonstrate the underlying common trend across all questions without accounting for deviations between questions.

The mean for the rate parameter is 0.86, (HDI 0.85-0.87), confirming the credible interval does not include diagonal lines, and the hypothesis stating that the growth rate is linear is rejected. The rejection of the linear hypothesis is also rejected under the error simulation process in 10 out of 10 simulations. Thus, the rate of idea generation over time appears to decrease exponentially as a function of the order in which the instance was received. This supports the intuition that workers generate overlapping sets of ideas, and in turn motivates deconstructing the influences on this rate decrease and the degree to which they can be controlled to achieve closer to ideal performance.

Note that this model does not accurately describe the process of idea generation. Sampling from the model would produce a cloud of disconnected points around the fit line, rather than a set of connected campaign lines.

5.3.2 Decaying Bernoulli model

The exponential model establishes a belief that the rate of idea generation is non-linear. However, it does not accurately represent the process of idea generation. First, it represents idea generation as a continuous process where, at any point in time, the number of ideas sampled from the distribution can be non-integer or negative. Second, by encoding the random noise with a normal variable, idea gathering is represented as sampling a unified mass of ideas at each point in time, rather than consecutive individual ideas as occurs in the real world. Furthermore, in the case where more than one person participates in brainstorming, it is expected that each person would have their own effect on rate generation. The exponential model does not adapt to this reality, as introducing per-participant rate parameters would result in predictions of the cumulative idea count assuming that all previous instances had been provided by the same participant. Thus, while the finding of non-linear growth is valuable, a new model is needed with greater descriptive power and improved accuracy.

Based on the previous model, exponential decay is adopted as an assumption and the requirement that a model be able to encode a linear relationship is dropped. Furthermore, it is desirable to explicitly model rate instead of the cumulative number of ideas. This allows the model to encode the probability of generating a new response, such that a different probability can be employed for each new instance if desired. Finally, because the outcomes on the rate scale are discretely 0 or 1, it is possible to encode the random elements of the model with a Bernoulli variable rather than a normal:

$$\text{novel}_i \sim \text{Bernoulli}(\theta_i)$$

Where

$$\theta_i = r_{\min} + e^{\text{decay}*i} * (1 - r_{\min})$$

This new model is a *decaying Bernoulli model*. The probability of generating a novel idea at each point in time is described by exponential decay as a function of the number of responses already received. The outcome of interest is whether the ith instance generated is novel (not represented in instances already received).

The decaying Bernoulli model follows the intuitive understanding of the brainstorming process. At each step in the brainstorming campaign, the instance received is either *novel* (it is associated with an idea for which there is no previous example) or not (another instance of the idea has already been seen). The $r_{\min}$ parameter is minimum rate of

idea generation, to which the rate decays asymptotically. This parameter was added as it became clear that there was insufficient data in the corpora to see the rate of idea generation drop completely to 0. As with the previous model, uniform priors over conceivable data values are employed. To recover the expected cumulative number of ideas at instance j, $E[n_j]$, take the sum of the expected values of each of the Bernoulli variables:

$$E[n_j] = \sum_{i=1..j} E[\text{novel}_i] \tag{5.1}$$

$$= \sum_{i=1..j} \theta_i \tag{5.2}$$

The model was fit to the the data for each campaign, where each question asked and number of responses requested condition corresponded to a distinct campaign. The fit model is shown in Figure 5.3, converging in 3 chains after 3000 iterations each. In this case, the rate model is fit to the difference in the campaign lines at each instance, resulting in a sequence in which 1 represents a novel idea. The expected cumulative count is recovered as described to provide a graphical representation of model fit that is comparable to that of the exponential model. The model was fit using Stan, and the Stan language model specification is given in Appendix A.2.

The posterior rate parameter is -0.0103 (HDI -0.0118, -0.0087), which is within the posterior HDIs for the same model fit under error simulation in 10 out of 10 simulations. This is consistent with the assumption of exponential decay. Furthermore, because this model encodes a more accurate representation of the brainstorming process, it is expected that it better describes the data set.

To test this, the Bernoulli decay model was compared to the exponential model by combining the two in a mixture model with a mixing component λ over their liklihood. A full specification of the Stan-language mixture model is given in Appendix A.4. The posterior of the mixture component was 1.00 (HDI 1.00-1.00) in favour of the Bernoulli decay model. This HDI excludes 0.5. This implies the Bernoulli decay model better describes the data gathered. Thus, the model not only encodes a generative process for idea generation in line with intuition, but this model describes the data at least as well as the more general exponential model. The increased descriptive power will next be leveraged to examine the extend to which individual ability affects quantity outcomes.

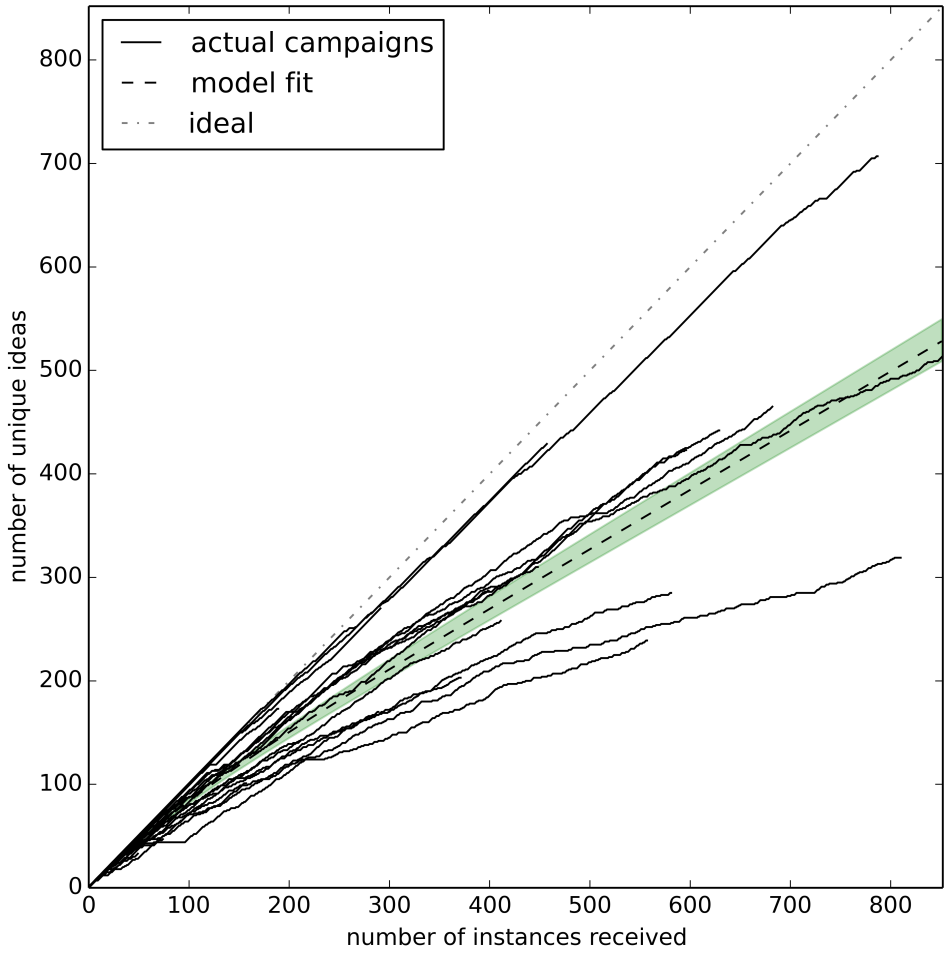


Figure 5.3: Bernoulli decay model fit: $\text{novel}_i \sim \text{Bernoulli}(\theta_i)$,
$\theta_i = r_{\min} + e^{\text{decay}*i} * (1 - r_{\min})$
The solid lines represent the actual number of unique ideas as instances were gathered. The dotted line is the fit model, while the shaded region demonstrates the 95% credible interval of the fit.

5.3.3 Decaying Bernoulli with participant parameters

While the improved descriptiveness of the decaying Bernoulli model is an advantage, its introduction was also motivated by the prospect of encoding the differences between participants. It is natural to expect that different individuals generate ideas at different rates. One individual attempting to game the system may produce 100 very similar ideas, while another may produce 100 distinct ideas. This variance in worker output is described explicitly in Chapter 6.

Participant effects are represented in a modification of the decaying Bernoulli model by introducing per-participant decay parameters:

$$\text{novel}_{ip} \sim \text{Bernoulli}(\theta_{ip})$$

Where

$$\theta_{ip} = r_{\min} + e^{\text{decay}_p * i} * (1 - r_{\min})$$

In this case, decay_p represents the decay of participant p's idea generation rate. The further below zero decay_p falls, the more rapidly the participant's idea generation rate decays towards the minimum. Intuitively, for each participant there is a different expectation of their ability to produce an idea that is novel to an existing corpus of N instances. A hyper-parameter is introduced on the decay parameters, reflecting the prior belief that the participants' idea generation abilities will be normally distributed:

$$\text{decay}_p \sim \mathcal{N}(\mu, \sigma)$$

This hyper-parameterization is necessary for the model to achieve convergence. As with the previous models, uniform priors over conceivable values are used for each parameter, with the exception of the participant rate parameters for which the normal distributions provides the prior.

The model was fit only to participants who generated 50 or greater ideas, as fewer responses were insufficient for the model to achieve convergence. The fit model is shown in Figure 5.4, for a subset of campaigns. This was done because a single fit for all campaigns has no meaning since each campaign has different participants. The model converged in 3 chains of 3000 iterations. The model was fit using Stan, and the Stan language model specification is given in Appendix A.3.

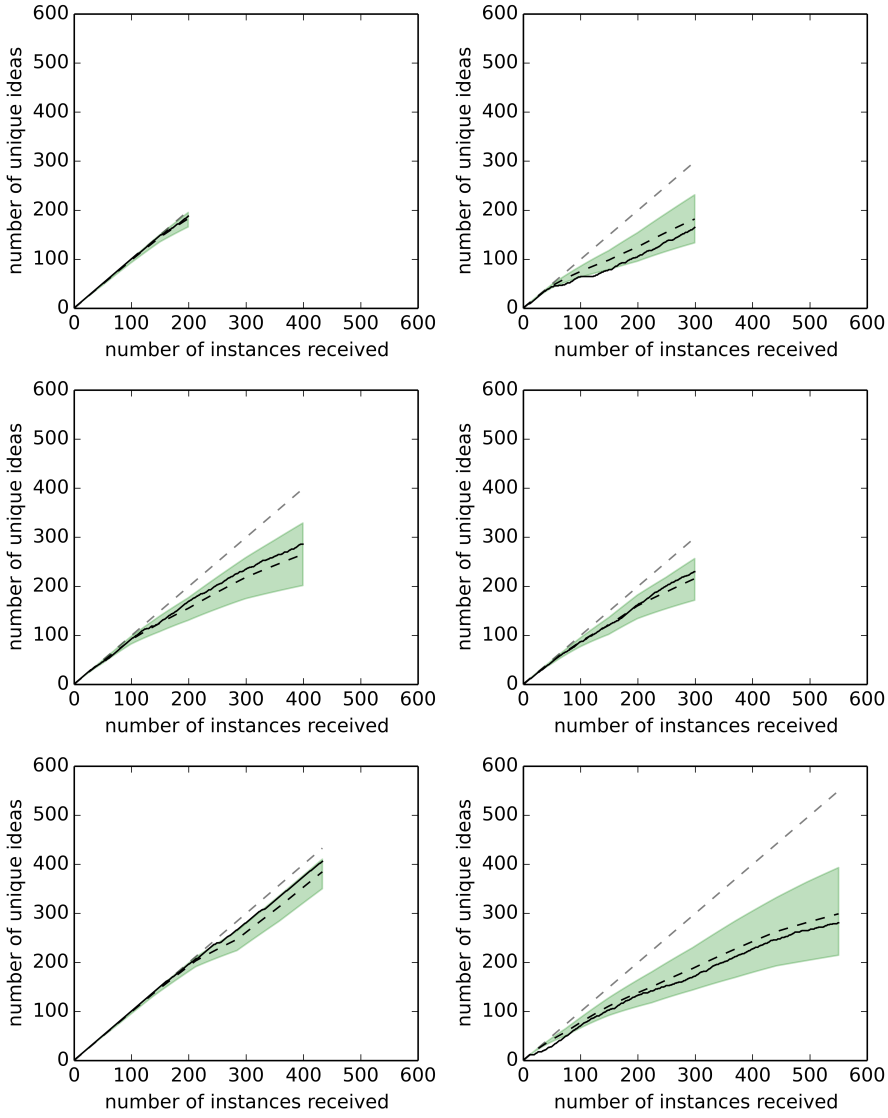


Figure 5.4: Bernoulli decay with participant parameters model fit:
$\text{novel}_{ip} \sim \text{Bernoulli}(\theta_{ip})$,
$\theta_{ip} = r_{\min} + e^{\text{decay}_p * i} * (1 - r_{\min})$
Given for six question and number-of-instances-requested combinations. The solid lines represent the actual number of unique ideas as instances were gathered. The dotted line is the fit model, while the shaded region demonstrates the 95% credible interval of the fit.

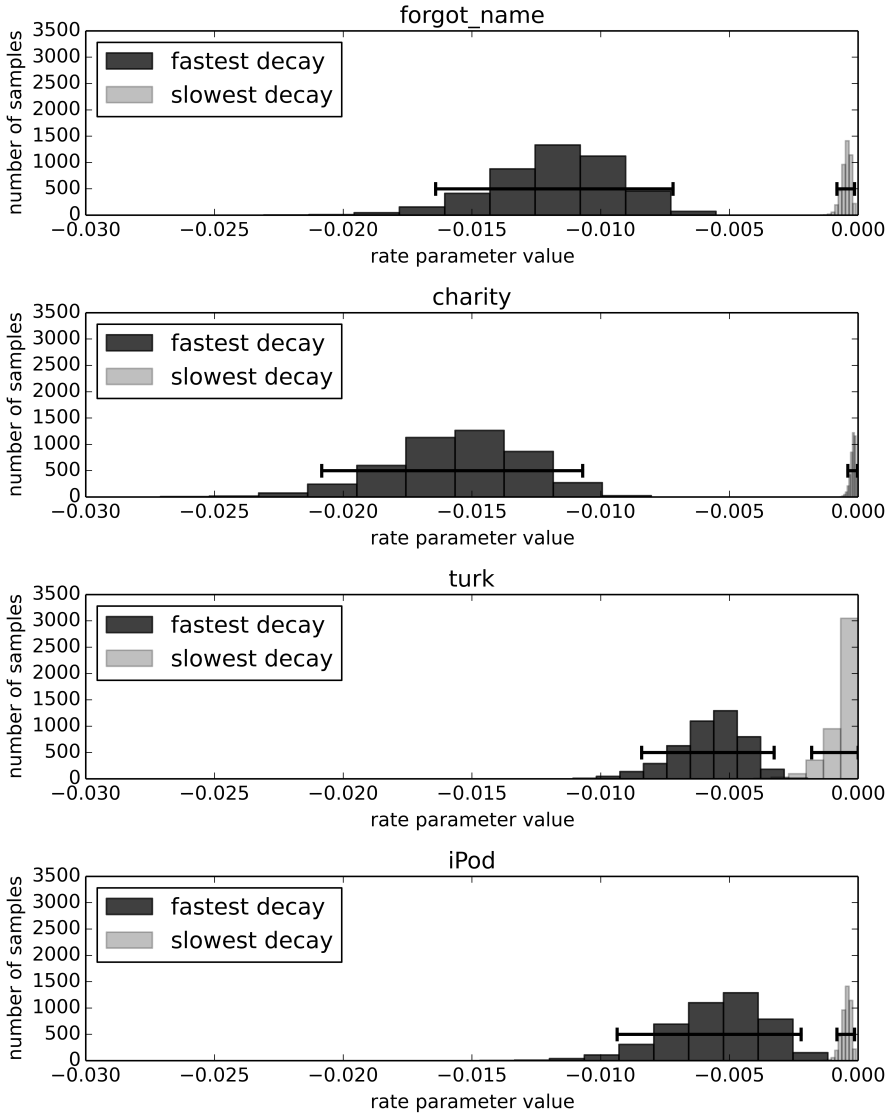


Figure 5.5: Posterior distributions for exponential decay parameters in the decaying Bernoulli model for the most productive and least productive workers. The bars represent the 95% HDIs for each distribution. The most productive user has a significantly larger decay constant, which results in an expected 34 more unique ideas generated out of 100.

In this model, the fit line is non-continuous (but still contiguous) - different participants "bump" or "flatten" the rate of idea generation as they contribute. While this model is less general - it is not expected to always receive participants with a similar distribution of decay parameters - by examining the posterior distributions of rate parameters, judgments can be made as to the distribution of "quality" brainstormers. Figure 5.5 plots the posterior distributions over the decay parameters for the most and least productive participants in each question condition.

As can be seen by the non-overlapping HDIs, the most productive participant has their rate of idea generation decay significantly less than the least productive participant. This means that variations in individual ability account for a significant portion of the variation in the number of ideas produced. In this case, the most productive participant would produce an expected 34 more novel ideas in a solo run of 100 instances than the least productive participant. This gap widens further to 63 additional novel ideas out of 100 when the same participants are contributing to a cumulative brainstorming pool that has already received 500 instances.

Note that this model fits an exponential decay curve for each participant based on a subset of that curve, dependent on the order of participants. This may lead to overestimation or underestimation of the decay rate for early or late workers, respectively. In the future work section, I describe some future models which may circumvent this issue.

These dramatic differences in number of ideas generated from worker to worker demonstrate that there are significant gains to made in performance. One approach is to create designs that push poorer workers towards higher performance. Another is to filter workers based on their ability to generate ideas. This thesis does not test these possibilities, but provides evidence that the current naive practices employed in crowd ideation are inefficient.

5.4 Idea novelty

Novelty was the second brainstorming outcome of interest defined above. In the introduction, one of the stated goals of this thesis was to understand *when* participants generate their most novel ideas. This has direct implications for the design of brainstorming tasks: it informs how many responses should be requested from workers, and the number of workers to recruit. Previous work by Parnes found that participants generated higher quality ideas in the second half of a brainstorming run [51]. This inspires the approach to novelty examination in this section: novelty is examined as a function of order in a brainstorming run to identify similar phenomena.

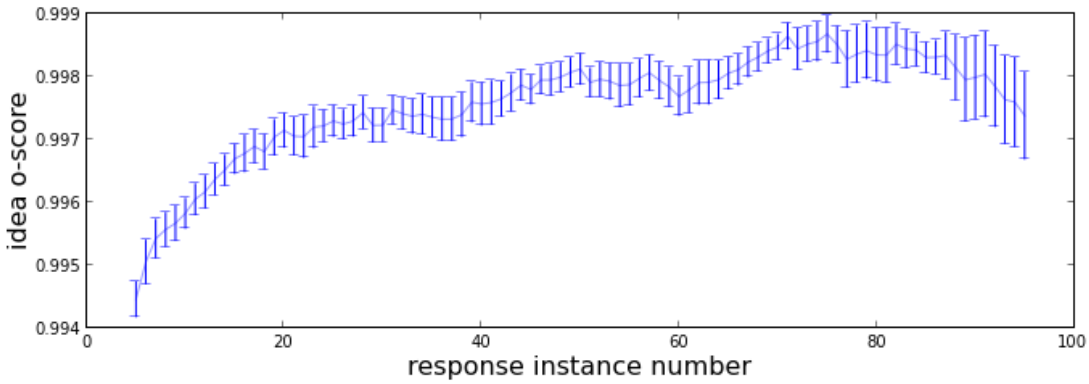


Figure 5.6: Mean idea o-score as a function of position in brainstorming run. Smoothed with a window size of 5.

Novelty is quantified as o-score, a measure of the prominence of an idea relative to the size of the data set. The o-score is defined on (0, 1) with a higher score corresponding to a more novel idea or category. Figure 5.6 shows the mean and standard error of idea o-scores as a function of position in brainstorming runs across all corpora, with an averaging window of width 5 on the order dimension for smoothing.

There are two elements of immediate interest in the figure. The first is the high lower bound on mean o-score, with scores all above 0.99. This indicates the high number of distinct ideas in the dataset. As suggested by the idea forest visualizations in Chapter 4, there are a huge quantity of singleton or near-singleton category trees which make up a negligible portion of the idea mass of the forest. Since the o-scores subtracted from one must sum to one, the quantity of ideas pushes the o-scores into this constrained range.

The second element of note is the clear upward trend: as they proceed through their brainstorming run, participants generate more novel ideas — ideas that are less common in the idea forest. However, this increase seems to plateau after a certain point. In this section, this plateau is envisioned as a symptom of a *change in strategy* by participants over time as they were forced out of their comfort zone. To capture this belief in shifting outcomes, I applied a *mixture model*, in which one model for the expected novelty of ideas gradually gives way to another. I found the distribution of o-scores at each point in a run was best described by beta distributions. Thus, two random beta variables are introduced from which an o-score is sampled. The mixture weighting of these models is a function of the position in the brainstorming run. This mixture model is defined as:

$$oscore \sim beta((1 - h(x) * \alpha_1 + h(x) * \alpha_2, \tag{5.3}$$
$$(1 - h(x) * \beta_1 + h(x) * \beta_2) \tag{5.4}$$

Where $h(x)$ is the function describing the mixing between the two models. Based in the belief that the o-score grows over time before plateauing, a naive mixing function entails increasing the mixture component linearly over time towards the second model until it completely replaces the first:

$$h(x) = \begin{cases} x/s & : x < s \\ 1 & : x \geq s \end{cases}$$

Where s is a model parameter representing the point at which the second model completely defines the distribution of o-scores, and is a parameter in the model to be fit. A uniform prior over (1, 100) was used for the s parameter, while the beta distributions were re-parameterized in terms of the mean and total prior count, as described in Gelman et al. [23].

This model was fit using Stan (the full model specification in Stan language is given in Appendix A.5). The resulting model converged in 3 chains in 6000 iterations. The fit is given in Figure 5.7

The resulting mean for the s parameter (the point at which idea stop increasing in novelty) was 18. The HDI was (14, 24), the bounds of which include neither 0 nor 100, suggesting that the distribution of novelty is better described by a mixture model than a constant. Furthermore, the second model produces significantly more novel ideas, suggesting that the novelty of ideas does increase over time. This result is surprising in that it suggests that participants do not run out of novel ideas, but rather run out of common ideas after which they reach a period of extended novelty.

This split point found under error simulation falls within the (14 , 24) HDI in 8 of 10 simulations.

As a result, I am able to present an empirically-derived guideline for those performing brainstorming tasks on microtask marketplaces: to receive the most novel ideas, ask participants for at least 19 responses.

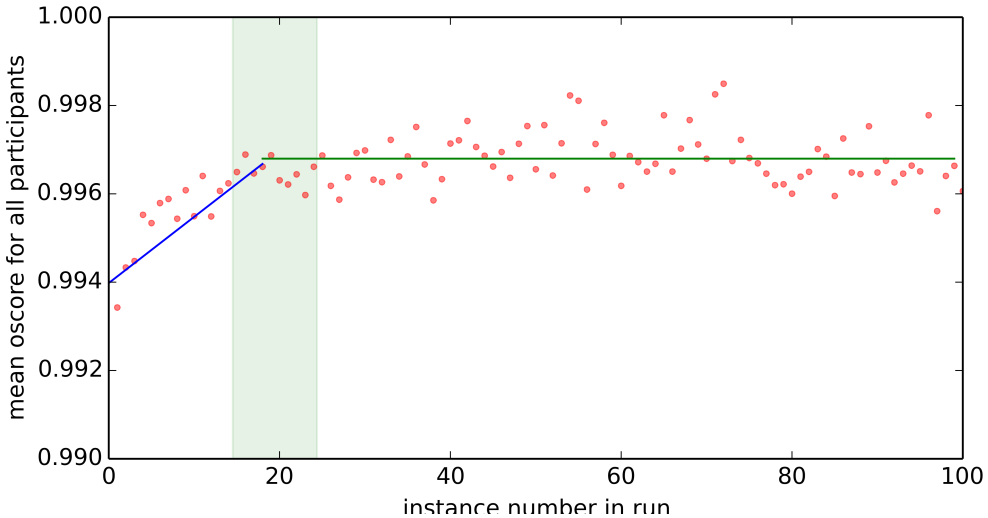


Figure 5.7: Fit of idea novelty mixture model. The shaded region represents the HDI for the split parameter

5.5 SIAM Replication

In Chapter 2, I made reference to the SIAM model by Nijstad and Stroebe [47]. Under this model, idea generation involves the repeated activation of images, from which several ideas are generated in sequence. These images were compared to the category trees of the idea forest, and each idea generated to instances in the sequence of brainstorming runs. In this section, that relationship is explored by examining two hypotheses from the SIAM model and their applicability in the context of microtask marketplaces. This replication provides a first demonstration of the similarities and differences between traditional and crowd brainstorming. While each hypothesis holds, they vary in effect size.

5.5.1 Category changes

SIAM suggests that individuals will generate ideas from an idea category until they exhaust that category, at which point they will switch to another category. This is re-expressed as the hypothesis that an idea from one category should be more likely to be followed by an idea from the same category than would be expected if ideas were generated independently of preceding ideas. The corresponding null hypothesis is that an idea in the same category

is no more likely to be generated than would be expected by random chance. Category changes can be detected in the context of the idea forest — two consecutive ideas in a run that have no path between them in the idea forest (i.e. are in different trees) represent a category change.

I model the probability of this category change with a Bernoulli random variable:

$$s \sim \text{Bernoulli}(\theta)$$

Where s is defined for each sequential pair of instances as a success if they belong to the same category tree, and θ is the success probability parameter. A uninformed beta prior is used for θ. It should be noted that there is a minor violation of an independence assumption in the Bernoulli sampling. It might be expected that a participant already in a chain of riffs on the same idea would be more or less likely to give an idea in the same category than someone giving an idea for the first time. However, this violation is not addressed in Nijstad and Stroebe's model, so independence is assumed.

If this credible interval for θ does not contain the probability of two consecutive instances in the same category assuming random chance, then the null hypothesis can be rejected. To determine the probability of two consecutive instance categories c_i, c_{i+1} being the same by chance, two simplifications are made. First, $p(c_i = a|c_j = a) = p(c_i = a)$. This follows from the specification of random chance; all instances are sampled independently. Also following from this specification is the simplification $p(c_i = a) = p(c_j = a)$, as if instances are generated by random chance then order has no effect. Thus, the probability of the same category occurring in consecutive instances is:

$$\begin{aligned}
p(c_i = c_{i+1}) &= \sum_{a \in \text{categories}} p(c_{i+1} = a, c_i = a) && (5.5)\\
&= \sum_{a \in \text{categories}} p(c_i = a)p(c_{i+1} = a|c_i = a) && (5.6)\\
&= \sum_{a \in \text{categories}} p(c_i = a)p(c_{i+1} = a) \text{ (by the first simplification)} && (5.7)\\
&= \sum_{a \in \text{categories}} p(c_i = a)^2 \text{ (by the second simplification)} && (5.8)
\end{aligned}$$

Fitting this model with an uniform beta prior (using an analytical solution rather than a sampling approach), the posterior mean for θ is 0.21 (HDI 0.20-0.22). For the

brainstorming corpus across all questions, $p(c_i = c_{i+1}) = 0.03$. This is well below the lower bound of the θ HDI, allowing the rejection of the null hypothesis that category-following is no more likely than would be explained by random chance. This finding held in 10/10 error simulations. This is consistent with the findings of Nijstad and Stroebe, and supports the concept that individuals work within categories of connected ideas and do not generate uniformly random ideas. This finding supports the idea that the SIAM cognitive model of alternative image activation and idea generation phases applies in microtask marketplaces as well.

5.5.2 Idea generation time

The second hypothesis presented by Nijstad and Stroebe is that the time it takes to generate an idea should be longer when changing semantic categories than when generating ideas within a category. To test this, I modeled the distributions of time spent for category-changing and within-category instance generation.

The time spent to generate an instance in brainstorming runs was experimentally determined to be well described by a log-normal distribution. Though it makes intuitive sense that participants would take more time to generate ideas later in a run, no observational evidence was found for this effect, and so order of generation was not accounted for in the models. Instead, a simple log-normal model was fit for idea-generation time:

$$t_c \sim lognormal(\mu_c, \sigma)$$

Where $c \in \{\text{between-category}, \text{within-category}\}$, t_c is the time to generate an instance in the corresponding condition, μ_c is the mean of the log-normal for the corresponding condition, and finally σ is the standard deviation of the log-normal. As usual for unbounded parameters, a uniform prior over credible values was used.

The model was fit for both within-category and between-category consecutive instances, the result of which is given in Figure 5.8. The full model specification is given in Appendix A.6. Sampling converged across 3 chains in 1500 iterations. The mean for time within-category ($\mu_{\text{within-category}}$) was 9.81 (HDI 9.75-9.88). The mean for time between-category ($\mu_{\text{between-category}}$) was 10.36 (HDI 10.36-10.42). As visible in Figure 5.8, these HDIs are non-overlapping, with between-category idea generation taking significantly more time. This relationship holds in 10 of 10 tests under error simulation. Between-category instance generation took an average of 0.57 seconds longer than within-category. This is inconsistent with the Nijstad and Stoebe finding of 6-12 seconds. This suggests that some property of

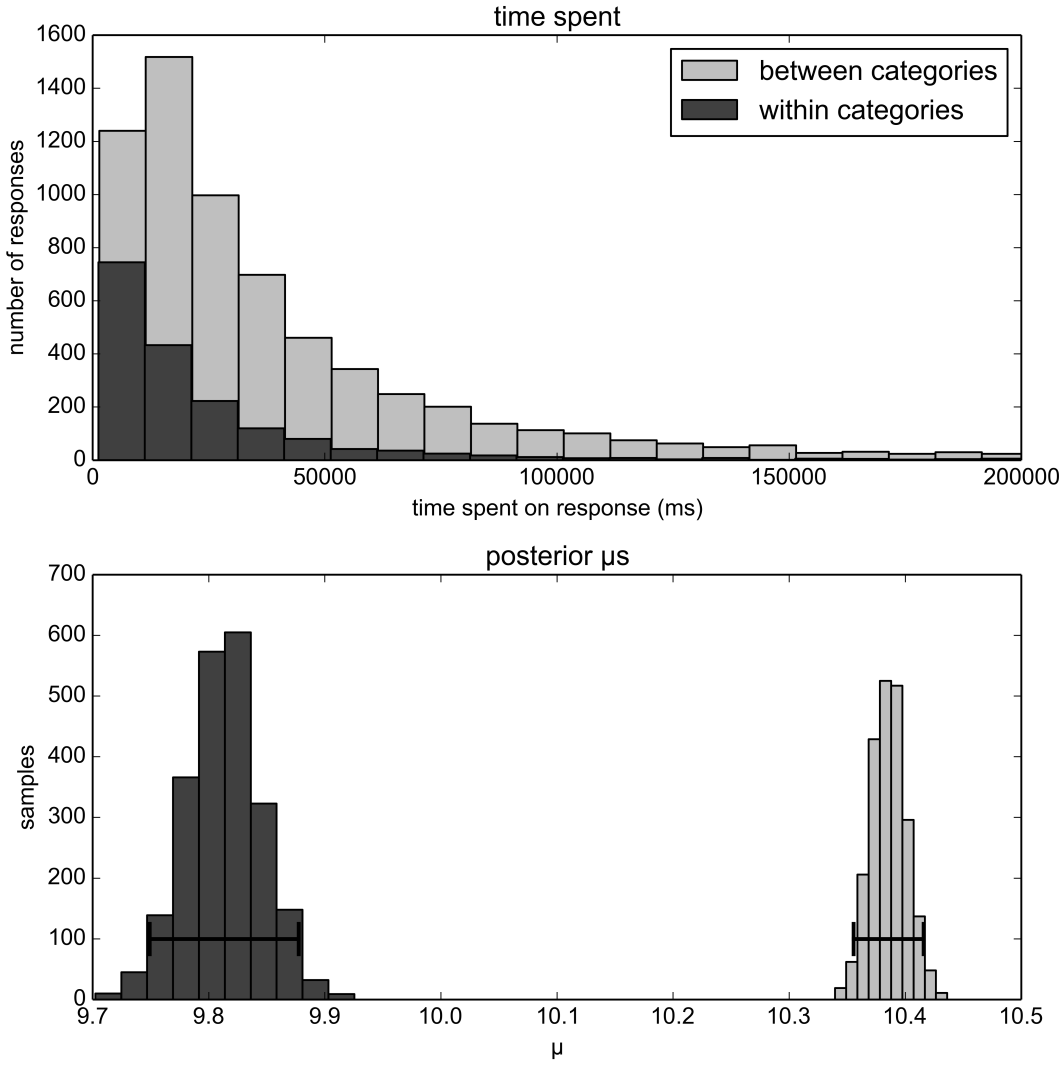


Figure 5.8: Fit of idea generation time for between-category and within-category idea generation. The upper panel is the histogram of data and the resulting log-normal distributions. The lower panel shows the posterior sampling distribution of the μ_c parameters.

brainstorming in microtask marketplaces either reduces the impact of category-switching or increases the cost of generating each idea. Further work is necessary to distill these influences.

5.6 Between-question comparison

The models described above are baselines for comparison, and were fit with all of the data available from each question gathered. A question of primary interest is how these models react to differences in the question asked. Guidelines established for the rate and novelty of brainstorming sessions are useful, but must be informed by influences that result from the questions. In this section, I briefly examine the rate model in the context of the different question conditions.

Figure 5.9 is the fit of the Bernoulli decay model for each question, with the shaded regions representing HDIs. All parameter priors, chain and iteration settings were the same as the original model, except that the posterior parameter means for the all-question fit were used as parameter initializations for the individual question models to speed up convergence.

The model fit demonstrates the differences in quantity outcomes between different brainstorming prompts. The charity campaign results in significantly more unique ideas as a function of instances, while the forgot name campaign results in significantly fewer. Despite this, the high-level trends identified across-questions are maintained within each question. Specifically, each question campaign is subject to decay and does not grow linearly, and as shown in Figure 5.5, each question shows significant differences in participant performance.

It is far more difficult to disambiguate the reasons for these differences in quantitative performance. What differentiates one question from another? Zagona et al. [71] propose three kinds of brainstorming tasks. Do these provide an adequate causal explanation for difference in performance? The problem of deconstructing brainstorming questions, and the causal link between these deconstructions and quantitative performance is beyond the scope of this thesis. What I have established in this section is reasonable evidence of two things.

First, that the models proposed produce similar high-level findings when applied to data from different brainstorming questions. This provides evidence for the generalizability of these models and suggests that they are representative of structures inherent to microtask marketplace brainstorming.

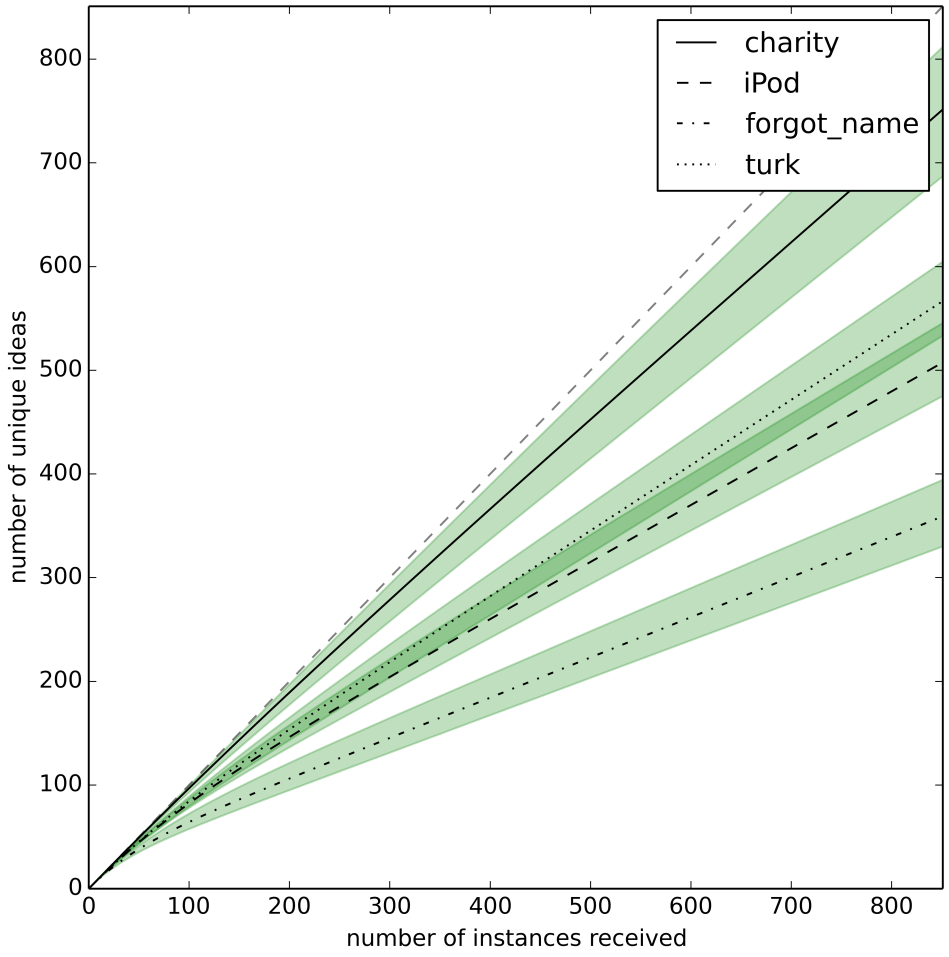


Figure 5.9: Fit of Bernoulli decay model for rate of idea generation, between questions. The lines represent the cumulative idea count as a function of number of responses received for the fit model. The shaded regions represent the HDIs for each question.

Second, that these high-granularity similarities are complemented with parameter-granularity differences that are significant. These differences may be grounded in a qualitative deconstruction of the questions, but further work is needed.

5.7 Summary and Discussion

In this chapter, I established three models for the rate of idea generation. The fits of these models to the brainstorming corpus contribute to solving the open problems in brainstorming task design. In particular, the Bernoulli model with exponential decay provides evidence that the rate of idea generation is non-linear, and that individual variation in brainstorming ability is significant and results in dramatic differences in the quantity of generated ideas. This distinction in individual ability motivates future work to identify the abilities of potential brainstormers early to maximize output, either by filtering workers or designing interventions. This model was also applied to make comparison between questions, and though the high-level properties of the model fits remain the same, there are significant differences in brainstorming outcomes.

A model of the novelty of ideas generated over the course of a brainstorming run was introduced, founded on the idea that participants shift between two strategies of brainstorming. It was found that participants shift fully to the second, more novel mode after their first 18 instances. This provides a simple design guideline for use by both practitioners and researchers.

Finally, models were constructed to examine two of the hypotheses proposed by Nijstad and Stroebe [47] in the context of microtask marketplaces. It was found that both held; participants were more likely to generate two ideas from the same category than they would by random chance, and it took longer to generate ideas between-categories than within-categories.

While the statistical inference in this chapter is a major contribution of this thesis, the models themselves are also of value. As demonstrated in the decaying Bernoulli model with participant parameters, these models can be used to make explicit statistical tests. In this thesis, a parameterization was introduced to explore the difference in rate between participants. Future research that introduced interventions or interactivity to a crowd brainstorming task could similarly utilize these models to test how those changes impact quantity, rate and novelty. For example, asking a worker to take breaks might increase the minimum rate of idea generation, while an explicit training session could decrease the rate of decay for the least novel participants.

These quantitative results aid understanding of how brainstorming behaves in a microtask marketplace. However, understanding *why* these are the properties of crowd brainstorming requires a qualitative understanding of how participants brainstorm. The next chapter will introduce a taxonomy of brainstorming strategies that provide a first step in this process.

Chapter 6

Qualitative strategies of idea generation

6.1 Introduction

In addition to the measures of quantity and novelty, I am also interested in what makes brainstorming responses qualitatively different from one another. In the past, research has focused on metrics such as realism and practicality. In this chapter, I focus more on the problem of idea generation as a process. An understanding of how an idea is generated in addition to what makes it more or less desirable can inform the design of brainstorming tasks. In addition, if different processes for generating ideas manifest in different artifacts, then by understanding these artifacts there is a potential for interactive brainstorming, in which a system responds to the worker's input of ideas.

This chapter describes a qualitative analysis of the brainstorming corpus established in Chapter 3. Open coding methods are applied to identify trends in brainstorming runs. The result of this coding is a taxonomy of *strategies*, plans or rules that are used by participants to generate brainstorming responses. Examples of each strategy are given, as well as descriptive statistics for the prevalence of these strategies in a random sampling of the data set. This chapter closes by discussing potential applications for the strategy taxonomy.

6.2 Coding for strategies

The quantitative models in the previous chapter form the backbone of this thesis. However, it is likely that the extensive corpus of brainstorming responses also includes valuable qualitative information. For example, before any qualitative coding had begun, it was hypothesized that it might be possible to identify *personas* of participants, such as the "Eager beaver" and "Lazy turker" personas identified by Bernstein et al. [6]. Qualitative labels might then be related to the quantitative outcomes of participants' runs. Obtaining any qualitative understanding of the brainstorming data was a difficult challenge due to the remote nature of microtask marketplaces. Participants could not be directly observed. Instead, a data-driven open coding approach was taken.

I selected the iPod data set to code because it represented the largest set of participants and was, at the time, the only corpus of brainstorming responses that had been encoded as an idea forest. I selected 30 random brainstorming runs (5 from each task length condition). Entire runs were sampled rather than individual instances because they provide information regarding the temporal relationships between instances. The results of the within-run novelty model and the SIAM replications in Chapter 5 suggest that this temporal ordering has influence on the kinds of ideas generated and their novelty.

As an initial coding strategy, I separated the 30 runs into categories based on subjectively evaluated metrics including length of instances, linguistic complexity of instances, creativity of instances, and prevalence of riffing. However, no clear personas of brainstormers emerged from these categorizations, attempts to establish personas resulted in large overlaps, and the categorizations themselves changed dramatically when different runs were sampled and re-coded. Furthermore, metrics based on the relationships between small subsets of runs (such as riffing) were providing the justification for discriminating between runs, instead of qualities derived from the holistic run. Based on this intuition that the discriminating qualitative factors between runs were based on factors larger than individual instances but smaller than the holistic run, the goal of attaining personas was abandoned.

Instead, a new open coding was performed that emphasized the relationships between instances. Each run was examined in temporal order, and labels were assigned as relationships between instances were noticed. This process was repeated iteratively until no new relationships were identified. For example, one of the early iterations focused on identifying instances which were riffs on previous instances in the run. A later iteration identified a distinctions between responses that utilized specific details of the brainstorming prompt versus those that solved more general problems. In subsequent iterations, relationships

that had already been identified were not coded for again, in favour of identifying new relationships that were parallel and distinct. Finally, after several iterations in which no new strategies were identified, another 30 randomly selected runs (5 from each condition, 1150 instances total) were selected and coded in an identical iterative fashion. No new relationships were established with this second set of runs. Each of the final identified set of instance relationships was re-defined as a *strategy*, a plan or rule that would produce relationships of that type.

6.3 Strategy taxonomy

A *strategy* is a plan or rule, that is used by a participant to generate one or more brainstorming responses. I identified three strategy categories, each of which can be employed with or without any of the other strategies in the course of a brainstorming run:

1. *Problem scoping* is a process in which a participant chooses a limited implication or component of the problem and provides multiple solutions.
2. *Riffing* is similar to the quantitatively established measure of idea re-use described in Chapter 3, but also qualitatively apparent in the data.
3. The *partial solution* strategy entails creating an idea that requires further idea generation to meet the requirements of the brainstorming prompt.

Each strategy is discussed first at a high level, and then decomposed into various substrategies which are variations on the primary theme.

6.3.1 Problem scoping

Problem scoping is a transformation of the brainstorming problem into a new problem. Solutions are generated which are applicable regardless or in spite of details given in the original problem statement. A rule of thumb for identifying problem scoping is to consider how much the details of the problem specification could change such that the instance is still a valid solution.

For example, many transformations of the iPod problem were observed. These include: brainstorm uses for *an audio output device*; brainstorm uses for *a hard drive*; and so on. A selection of responses from Participant 270 (P270) can be used to demonstrate the concept:

- have them as a resource on public transportation – people must supply their own headphones
- use them in museums to give information on various installations
- have cities install them in tourist areas, so people can listen about where they are

The participant's responses satisfy a re-scoped version of the original problem: "brainstorm uses for an audio output device". Any instance which provides a solution to the transformed problem also provides a solution for the original problem.

Different transformations of a problem define a continuum of corresponding solutions. All solutions rely on some subset of the details in the problem. However, two levels of problem scoping were particularly prelevant in the iPod data set. The first is *focus scoping*. Focus scoping transformations reduce the scope of the problem by focussing on a single detail from the original problem statement. The previous responses, which respond to a question where the only remaining detail is audio output, are an example of focus scoping. Responses generated by focus scoping are applicable to the transformed problem, but not applicable to any problem which changes the critical detail of interest.

The other common type of scoping is *defocus scoping*. Defocus scoping involves transforming a problem by stripping *explicitly-stated* details to solve a more general class of problem. Consider the original problem statement of the iPod question:

> Many people have old iPods or MP3 players that they no longer use. Please brainstorm N uses for old iPods/MP3 players. Assume that the devices' batteries no longer work, though they can be powered via external power sources. Also be aware that devices may *not* have displays. Be as specific as possible in your descriptions.

The explicit details of this problem are:

- the device provides audio output
- the device needs external power
- the device may or may not have a display

A defocus problem scoping would encode none of these details. An example of a defocus transformation of the iPod problem is "brainstorm uses for a physical object". For example (P301):

- doorstop
- use in abstract art

Virtually any stated detail of the original problem could be changed while still maintaining the applicability of the above solutions. For example, the iPod could be replaced with an old pair of shoes, audio playback could be replaced with the ability to summon elephants from thin air, and it doesn't matter that the object has functionality that is enabled by a supply of electricity.

Some responses employ no scoping whatsoever. These responses utilize the device exactly as it was intended, as a portable MP3 player. For example, "get it fixed and give it to a needy kid" (P89).

6.3.2 Riffing

Riffing is a strategy in which a new instance is generated as a manipulation of an instance earlier in the brainstorming run. These ideas need not be consecutive, and in fact often are not. I identified four ways in which riffing was generally manifested: generalization riffing, repeat riffing, hold riffing, and continuation riffing.

Generalization riffing

Generalization riffing occurs when a participant generates two or more ideas and one is a generalization of the other. For example, two consecutive ideas given by P130:

- brick
- building material

In this case, iPods could be used as bricks in construction, but the participant then expands upon this concept by suggesting that iPods could substitute for a larger class of building materials. He or she provides a second answer which encompasses the first. Of course, generalization riffing can also occur in the opposite direction, in which the turker begins with a general concept and then provides specific instantiations.

Repeat riffing

Repeat riffing describes a process under which a participant exhaustively identifies permutations of a response by replacing a limited portion of the response, generally a single term. These replacements could all be summarized by a single concept. For example, some responses from P151:

- extract usage data
- extract texts
- extract gps data

In this case, all three ideas could be summarized with "extract data". The object of the idea statement is exhaustively substituted with items from a restricted category. Identifying repeat riffing requires some subjective evaluation to determine whether the instantiations add information or simply enumerate the obvious shape of an implied solution space.

Hold riffing

Hold riffing is a strategy in which participants hold at least one element of a previous response constant when generating a new response. Unlike repeat riffing, hold riffing examples always introduce additional information not encoded in the source idea, and the portion of the response held constant may not be summarized by repeated phrasing. For example (P130):

- Jukebox music selector in bars
- Commercials in bar bathrooms
- Tapper handles for beer

In this example, the setting of the application is held constant between all ideas: a bar. Another realization of hold riffing holds language terms or phrasing constant between unrelated ideas. For example (P230):

- We could use the hard-drives inside for different electronics.

- We could use them in place of rocks (to throw at things, to use in pavement.)

In this example, the participant may be using the common phrase "we could use" as a prompt to more easily flow into an idea. Notably, while the repeated phrasing is similar to that identified in repeat riffing, the second response provides a solution that is distinct from the first, as using the device hard drive does not naturally imply using the device as a substitute for a rock.

Continuation riffing

Continuation riffing is the strategy of creating another idea that cannot be understood without the context of the previous. For example (P252):

- I suppose you could just grind them down into a sand
- You could take the sand... and put it in an hourglass

Without the context of the first idea, it is unclear what the sand in the object of the second idea phrase is. This type of riffing is exclusive to ideas that encode an explicit plan of action, as in the responses that could be decomposed into "steps" in Taylor et al. [62]. Continuation riffing is often related to the partial solution strategy described below, as a set of continuation riffs may encode an entire solution when each instance alone does not.

Spatial separation

In the process of identifying riffing strategies, I encountered some surprising characteristics. I expected most riffing to occur between consecutive instances, where a riff on an idea directly followed the source idea. I call this kind of riffing *consecutive riffing*.

However, many riffs came further away in the run from their source idea. I call these riffs *reach-backs*. Reach-backs occurred throughout a run, with participants as likely to riff on an old idea as a more recent one.

Finally, some participants *pair* their riffs. Paired riffing is a special case of consecutive riffing in which only a single riffed idea is produced. For example, paired repeat riffing from P276:

- cut up and use to decorate shoes

- cut up and use to decorate vase

I comment on this phenomenon in particular because of its surprising frequency. Some participants would riff almost exclusively in pairs. In Chapter 3, the median length of a chain of riffs derived from idea forests was 2, supporting this finding. It remains an open question why participants are willing to provide derivative versions of their own responses in small groups, but do not further exploit this strategy to minimize time spent and thus maximize reward.

6.3.3 Partial solutions

Under the *partial solution* category of strategies, responses provide some elements of a solution to the problem, but further idea generation is required to implement the solution. The following ideas (P277) are all examples of partial solutions:

- use old parts to make a new device
- old parts can use to make something
- create a new device

In these responses, there is a recommendation that a goal of solutions be to produce new electronics, but it is unclear what would be made, and how it would be made. Four partial solution sub-strategies were identified: problem scoping without solution, goal without plan, plan without goal, and passing the problem.

Goals and plans

In many cases, a problem solution can be broken down into two components: a goal that is to be achieved, and a plan of action to achieve that goal. In the context of the iPod problem, goals are end-uses for the old iPod or MP3 player, while a plan of action would be a sequence of steps to transform the old broken hardware into something that could fulfill that end-use.

In some responses, an end goal has been established, but executing the goal requires additional information. Essentially, there is no plan of action to achieve the goal. We call

this strategy *goal without plan*. The responses that begin the partial solution section are examples of the goal without plan strategy.

Conversely, the *plan without goal* strategy gives an action to be taken without establishing the purpose or end goal of that action. Plan without goal is demonstrated in these responses (also from P277):

- melt down old parts
- see what old parts are usable

In this case, the operations of melting down the device or checking for working parts can be performed, but don't provide any obvious benefit. There is a plan of action, but no goal.

Responses that do not have an explicit goal or plan may nonetheless imply one. For example, "throw it in the ocean" (P89) has the implied goal of disposing of the broken device. Similarly, an idea can have an implied plan, as in the answer "use it as a doorstop" (P259), in which no transformation of the device is necessary. These cases are subjectively evaluated by the coder based on the ability to easily and naturally justify the plan, or fill in the plan for a goal.

Problem scoping without solution

Above, I described the problem scoping strategy, in which participants provide solutions to a transformation of the original problem. Occasionally, the transformation of the problem is the only information provided in a response. I call this strategy *problem scoping without solution*.

For example, a response to the iPod problem from P64: "Remove glass screen to make something". In this case, the participant has provided a scoping transformation of the problem ("What are uses for a small, rectangular piece of glass?") but has not actually provided a solution to the scoped problem. The transformation in itself is a useful lens through which to examine the problem, but if a requester was to receive only this response they could not implement it.

Pass the problem

Finally, responses in the *pass the problem* sub-strategy of partial solutions relocate the need for idea generation to a third party. These responses from P277 provide an example:

strategy	# instances	% of instances
focus scoping	267	23.2
defocus scoping	828	72
no scoping	55	4.8
hold riffing	278	24.2
continuation riffing	3	0.3
repeat riffing	119	10.3
generalization riffing	48	4.2
any riffing	448	39.0
pair	108	9.4
reach back	129	11.2
scoping without solution	3	0.3
passing the problem	69	6
plan without goal	58	5.0
goal without plan	58	5.0
any partial solution	188	16.3

Table 6.1: Prevalence of brainstorming strategies in sampled runs

- give to non-profit that can benefit
- give to an organization that has the knowledge to use these devices

In this case, a third party must determine what to do with broken MP3 players. Other examples of passing the problem could involve requesting the assistance of an expert or deferring the problem to a later time.

6.4 Strategy use

In this section, the prevalence of each of the identified strategies is discussed. The results of the second coding exercise of 30 runs (5 from each condition, 1150 instances total) are examined for prevalence of strategies and descriptive statistics are given. The prevalence of each strategy is shown in Figure 6.1, while Table 6.1 describes their prevalence as a function of the total number of instances in the set of runs.

Defocus scoping is by far the most common strategy, occurring in 72% of instances. This is particularly surprising, as it suggests that most of the ideas participants have for a

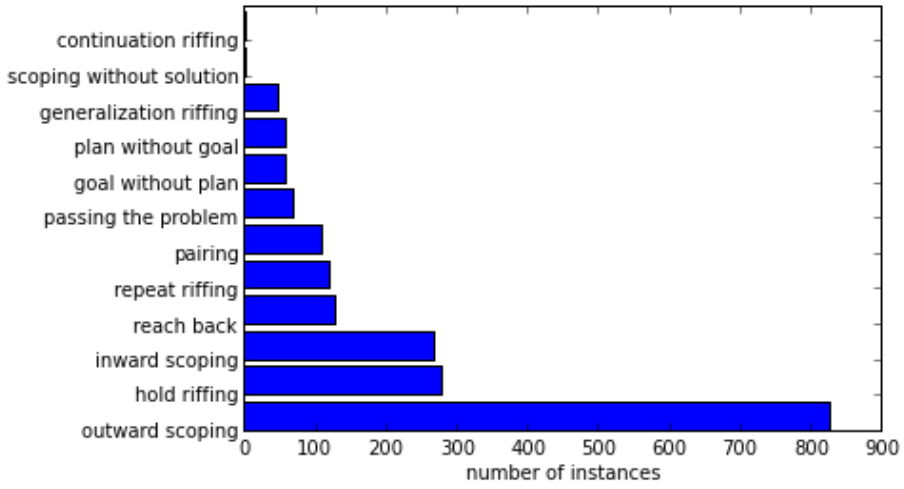


Figure 6.1: Presence of strategies

brainstorming prompt are not optimally suited to the problem. Alternatively, it suggests that when given a brainstorming problem, participants provide solutions not only to that problem but to a more general class of problems. This may enable the cross-applicability of responses to one brainstorming prompt to another. For example, a requester looking to brainstorm uses for an old pair of shoes may be able to request fewer responses to their direct problem and borrow defocus scoping solutions from the iPod problem.

Riffing is also very common, with 40% of instances employing some form of riffing. Riffs are most often consecutive, and hold riffing is the most common type. The prominence of hold riffing may be more of an effect of its definition than the prominence of a specific strategy; it functions as a catch-all for any riffs that do not fall under the other categories.

Partial solutions are fairly uncommon (16.3%), and most often involve missing either a goal or a plan to fully specify a solution (10% together). It is of interest to determine if partial solutions are a result of lack of effort on the part of the participant, or a lack of realization that the response is not fully implementable.

Continuation riffing and scoping without solution are the least common strategies, each with only 3 instances.

strategy	5	10	25	50	75	100
focus scoping	2	5	11	23	29	44
defocus scoping	3	3	8.5	27.5	39.5	41
no scoping	3	5	9	18	22.5	20.5
hold riffing	3	8	12.5	24	44.5	57.5
continuation riffing	—	—	18	4	27	—
repeat riffing	4	7	2	25	49.5	45
generalization riffing	4	4	11	29.5	34	30
pair	3.5	7.5	12.5	25.5	40	33.5
reach back	3.5	6	11	29.5	40	47.5
scoping without solution	—	—	—	13	—	53
passing the problem	1.5	—	14.5	40	35.5	60.5
plan without goal	—	—	6	18	38.5	30
goal without plan	—	3	2	15	11	29

Table 6.2: Median position of strategy in run as a function of number of ideas requested

6.4.1 Strategy location

It is also of interest to consider where in a brainstorming run the strategies are more likely to occur, as this could provide information to motivate on-line interventions based on any implied phases of brainstorming. Table 6.2 gives the median position of each strategy in brainstorming runs as a function of number of responses requested.

The distributions of strategy occurrence across position in run tend towards uniform. The notable exceptions are problem scoping and reach-backs. There are fewer occurrences of focus scoping strategy later in brainstorming runs. This suggests that participants run out of ideas that highlight a single component of the brainstorming problem, and are forced to employ defocus scoping. When considered in light of the Chapter 5 claim that participants generate their most novel ideas late in a brainstorming run, this suggests that this burst of novelty may be explained by a shift towards solutions to defocus scoped problems.

The amount of riffing increases with position. This suggests that participants are utilizing their old ideas to seed their new ones as they run out of the responses. However, these riffed responses are still highly novel (again as shown in the previous chapter), suggesting that participants riff late in their runs by "filling out" existing category trees.

These speculations need to be verified by future research before they are employed to generate design recommendations. However, they do suggest that the change in novelty

of responses may be in part explained by a change in the types of strategies employed, particularly a shift towards defocus scoping and increased riffing. If participants are in fact generating ideas later in their brainstorming runs that solve a different class of problems than those in the early parts of their runs, these differences must be examined in the context of how they impact the desireable outcomes of brainstorming.

6.5 Applications

Strategies provide an alternate means of measuring brainstorming outcomes. Discrete quantities of ideas are an ideal measure, but they require disambiguating semantic meaning, which is a process difficult to automate. Strategies may be more easily detected, particularly when they are the product of relationships between ideas. For example, detecting repeat riffing does not necessarily require a complex semantic understanding of ideas; edit distance may be a sufficient metric.

Furthermore, strategies may provide useful descriminants for comparing participants. As discussed in Section 5.3.3, participants vary in the novelty of their responses. Novelty measures like o-score cannot be evaluated on-line, or in isolation, as they rely on the collected response pool of many participants. In contrast, one can imagine detecting strategies as a worker brainstorms and intervening in the brainstorming process to promote or demote strategies.

Another potential application of this strategy taxonomy is to filter ideas for manual examination by requesters. Examining a large pool of ideas requires huge overhead, as evidenced by the extensive methodological considerations introduced in Chapter 4. If brainstorming runs could be automatically labeled with strategies, a filter could be constructed which returned only one response from each pair riff, or specifically selected non-partial solutions.

Finally, a taxonomy of brainstorming strategies allows task designers to proactively suggest strategies to participants. If certain strategies are found to produce higher quality responses, a brainstorming system could respond to pauses in productivity by suggesting one of those strategies. For example, forwarding an older response of the participant's and asking them to perform a hold riff operation.

6.6 Summary

This chapter presented a qualitative analysis of a corpus of responses to brainstorming questions. It identified strategies of idea generation that were evident in that corpus, and described the prominence and locations of those strategies within brainstorming runs. Finally, potential applications of brainstorming strategies were identified. In particular, strategies present an alternative to uniqueness-based measures of brainstorming runs, with the advantage that they can be evaluated and considered in an online environment without knowledge of other brainstormers, and are furthermore more tractable to classify automatically.

Chapter 7

Discussion

7.1 Introduction

Idea generation is a prerequisite to completing creative tasks. Thus, if as a research community we intend to use the crowd and microtask environments to complete creative tasks, we must solve the problems of idea generation in this environment. My own introduction to this problem came when designing a crowd workflow for a creative problem. Prior work has yet to generate set of guidelines or heuristics for designing idea generations tasks. In an ideal world, we could submit an algorithm a brainstorming question, a budget, and perhaps a handful of evaluation criteria, and receive a brainstorming task that returns a list of unique ideas ordered by quality. This algorithm would need to construct an ideal task, solicit workers to respond, coordinate and collect responses, and interpret these responses to extract pertinent metrics. However, there are many problems which must be solved before even a poor approximation of such an algorithm could exist:

- What is the design space of a brainstorming task?
- What is the design space of a brainstorming *prompt* (the question to which the worker must provide answers)?
- How many ideas should be requested from each worker?
- How many workers should be be asked for ideas?
- Who should be asked? How can appropriate brainstormers be identified?

- How much should workers be paid for this kind of task?
- What is the stopping criteria for gathering responses? Can this be evaluated automatically?
- What is the evaluation criteria for responses? Can this be evaluated automatically?
- What information should the task expose to the worker?
- Should the task respond to the worker? If so, how?

Of course, as these pragmatic questions are considered and addressed, theoretical questions inevitably join the fray:

- How do workers brainstorm differently in microtask marketplaces than other environments?

As is the habit of research questions, each of these could be explored in sufficient depth to compose a thesis. In this thesis, I have addressed some of these questions to a greater or lesser degree. For example, in Chapter 5, it was suggested that workers should be asked for at least 18 ideas apiece, and Chapter 3's idea forests provide a solution for a small part of the problem of evaluation criteria. However, where I have not been able to explore in depth, I have had the opportunity to briefly examine these questions in the context of a large brainstorming corpus. In this chapter, I will address a subset of these questions in the context of my experience.

7.2 Design space of brainstorming prompts

When iteratively designing questions for the study, there were a few properties which seemed to consistently improve response quality:

- *Adding constraints* reduces the space of possible responses and as a result may push brainstormers out of their comfort zone into genuinely novel territory more quickly. For example, the turk question was changed to specifically request features for a mobile app, which exclude simply improving payment. Providing more context, as with changing the charity question to refer to a specific institution, improves the validity of ideas (i.e. vastly reducing the proportion of "bake sale" responses).

- *Eliminating obvious possibilities* prevent them from dominating the results. For example, in the charity question, eliminating donation pages, merchandise, web advertising, and so on.
- It is useful for the worker if they have an obvious *evaluation function*, or way to assess the quality of a response by the same criteria the requester will use. Despite the tenet of deferred evaluation, brainstormers still self-filter, and for this reason it is better to have the evaluation function explicitly known. Often, this can be implied by the question itself. For example, in the forgot name question, it's self-evident that the evaluation function would be the strategy's chance of both learning the name and avoiding discovery. The turk question was modified to ask for app features that would "improve the worker's experience" — an evaluation function the workers are uniquely suited to predict.

In Chapter 5, I briefly discussed differences in the rate of idea generation between questions in the study. The forgot name question generated far fewer distinct ideas, while the charity question generated significantly more. In contrast, the number of categories generated per instance is fairly constant across conditions. I propose that these differences are explained by a phenomenon I call "decorating". In this sense, decorating is slightly permuting an idea such that the new idea encodes enough new semantic information to be distinct, but not enough that it belongs to an entirely different category.

In the case of the charity question, it is simple to vary an idea about selling merchandise to sell sweaters instead of shirts, or bumper stickers instead of mugs. I am not suggesting that participants decorate their own ideas in a form of riffing, but rather that certain responses have such a wide possibility space of decorations that participants are unlikely to overlap in the combinations they choose. In contrast, the forgot name ideas have fewer available decorations; there are fewer types of people you could ask for the subject's name in a social scenario than there are kinds of merchandise you could sell to raise money.

Despite this concept of decorations, the rate of purely orthogonal design elements (i.e. categories) seems constant between ideas. The rate of proposing new categories in brainstorming tasks may have fairly tight variance, even with the expected fluctuations from individual to individual. The decoration concept also explains the significantly different idea counts while allowing for a fixed expected overlap in undecorated ideas from participant to participant in spite of the prompt. Essentially, certain questions provoke responses that can have many distinct permutations, while other questions do not. I suspect that decorating can be identified in an early sampling of responses, and once a threshold is reached for decorations of a particular base idea, it may be valuable to re-issue the brainstorming task specifically prohibiting that idea.

7.3 Optimizing the number of ideas requested and workers recruited

In the course of this work, workers were asked for at most 100 ideas. It was surprising to find that even at this high value, there were not significant qualitative differences in the types of responses received. Furthermore, the more responses gathered per worker, the greater the proporties of unique ideas and categories. This rule held generally, but a few particularly high-productivity turkers in the 75 responses condition were able to raise the performance in that condition to dominance. The rule of thumb seems to be to ask for as many ideas as possible — it would be interesting to test if this can possibly hold in the degenerate case of gathering all responses from a single worker. Notably, there was an increase in early abandonment (submission of the task without completing all responses) in the 100 response condition, but it was not significant.

7.4 Identifying appropriate brainstormers

One of the prerequisites to proper brainstorming identified by Isaksen [28] is that the participants have sufficient expertise in the problem domain. The questions used in this thesis were selected in part because it was assumed that most workers would meet this expertise requirement. However, as the evaluation phase of this research began, it became clear that in the case of the charity question, the judges could not evaluate the expertise of the workers, for they themselves had insufficient experience coordinating charity efforts. In hindsight, it is clear that this problem is not unexpected but should in fact be common: if a requester has insufficient expertise to solve a problem, it is likely they also have insufficient expertise to evaluate solutions and the workers producing those solutions.

There are two approaches to solving this problem. The first is to evaluate workers in ways that do not require domain expertise. The second is to identify a source of domain expertise that can be used for evaluation. This latter approach is most common in previous work, in which expert judges are employed to rate ideas (and by proxy, workers), or additional crowd workers are solicited to provide rating by consensus. The problem with both of these methods of obtaining expertise is that they do not scale, requiring either employing expert judges or soliciting additional workers by an order of magnitude.

We are left with the first approach, which necessitates finding good evaluation metrics for worker quality without domain knowledge. This could be done by evaluating responses for reading comprehension level, searching for overlaps in vocabulary with expert sources

(for example, overlap with Wikipedia articles on the problem domain), and so on. One approach I feel is particularly promising is to look at how a participant's responses change over time with respect to various similarity metrics. Over the course of this thesis, it seemed that responders with a high *variety* in their responses also demonstrated significant expertise.

Finally, there is the potential to avoid the problem altogether. It is possible that requesters can leverage demographic and qualification information about workers to selectively provide tasks to appropriate workers. An expanded taxonomy of credentials and associated tests could be leveraged automatically by requesters, but as of the present resources are insufficient.

7.5 Payment

In this thesis, a rate of 3.5 cents per response was paid to workers. This value was selected to equate to $7 per hour at the mean rate of idea generation identified in early pilots. Lower values were experimented with in those pilots that were more closely tied to market values. While I performed no explicit tests on how this change in payment affected outcomes, it is notable that paying market rates resulted in several workers using the feedback field of the HIT to admonish the researchers for unfair pricing. I consider it likely that this general dissatisfaction led to a lower quality of response. These impacts need to be closely examined, and I caution against determining prices strictly as a function of market forces, which seem to drive towards exploitation in microtask marketplaces.

Despite the fixed per-idea reward, it seems turkers prefer HITs with high absolute value. The inverse statement, that turkers dislike low absolute value HITs, is also supported by experience in this study; fewer than 100 workers were willing to accept HITs to brainstorm five responses, with the rate of HIT acceptance dropping over time. This suggests that this study was able to very quickly exhaust the population of available workers (keeping in mind the population was limited to US residents). This is a problem which has vast implications for the long-term viability of microtask marketplaces. As more and more solutions are proposed which leverage crowd intelligence, it will become increasingly important that tasks are designed to be both competitive for attention and economically efficient.

7.6 Stopping criteria

In their straightforward form, the stopping criteria for a crowd brainstorming problem are simply when all available funds have been expended such that they maximize the number of ideas produced. However, I am particularly interested in the ways that problems can be solved when we consider not "what are the ideas we received", but rather, "what is the form of the idea *space*". Intuitively, when ideas are collected in a brainstorming task, each response is a sample of the space of possible ideas. By modeling this idea space, we can instead decide when to stop gathering based on how well we understand solutions to the problem. For example, collection could be stopped if all collected responses were co-located and restarted with another prompt. Or, ideas could be collected until there is sufficient information to begin generatively producing new ideas. Furthermore, it cannot be expected that the properties of idea spaces for different questions are the same. Thus, selecting a number of instances to request and strategies for requesting them *a priori* of any collection is unlikely to ensure a good representation of the space of ideas. Ten samples of an idea space containing five equally-represented ideas may be sufficient, but if there are ten or more ideas in the space, it is unlikely the sampling will given an accurate representation. This could lead to missing the great idea that would best solve the problem.

The expectation beginning this work was that it would be inexpensively possible to exhaust the idea space for a problem on Mechanical Turk. In the language of the rate models in Chapter 5, the hypothesis was that the rate of new ideas generation would decay asymptotically to zero, and reach zero within an affordable number of instances (around 1000 instances, or $35 at the rate paid). In contrast, it was found that while the rate of ideas did decay to an asymptote, this was non-zero. This suggests that given time, workers will generate an infinite number of ideas in response to a question. The reasons for this may be grounding in the problem scoping introduced in Chapter 6. If this infinite growth is related to the transforming of a question, it may be that responses to the specific question posed do converge more quickly. Identifying problem scoping and modeling the rate of non-scoped responses is a first step to teasing out this relationship.

Other kinds of convergence criteria may be viable with less work up front. For example, one could stop collecting ideas when there is a low probability than any "big" ideas (ideas that fill some minimum portion of the idea space) have been missed. For example, after collecting 100 instances, there is a 0.6% chance that an idea representing 5% of the idea space has been missed. This kind of thresholding approach only works if the outcome of interest is ensuring the identification of popular ideas.

Another approach would be to extract dimensions of solutions, and stop when the number of dimensions either converges or the rate of generation drops below some threshold.

A simple version of this would treat each vocabulary word as a dimension, and add dimensions when a word has been seen at least twice. It might be possible to use techniques such as Principle Component Analysis to further disambiguate dimensions that result from combinations of words.

7.7 Evaluation criteria

Measurement is by far the most difficult problem in researching creativity. Beyond the lack of a unified definition for the hybrid measure of creativity, obtaining scores for even its sub-components can be a challenge. For example, assessing utility requires predicting the outcome of a success function. When done with human judges, simple differences in interpretation have dramatic effects. A definition of utility which encompasses realisticness will produce a different score than one which does not. Similarly, a definition of realisticness can encode economic cost or can refer to adherence to the laws of physics. These are measures normally evaluated by judges, but access to expert judges is not a reasonable expectation for requesters on a microtask marketplace.

Several attempts at judge-based techniques for outcome scoring were attempted in the course of this thesis. The classic approach of rating ideas on ordinal scales was impractical. First, inter-rater reliability (IRR) was low, and discussion to resolve this reliability deficit often devolved into efforts to explicitly define ambiguous measures. Even had this method resolved into a reliable measure, I have doubts that the results could be replicated without an extensive training session conducted by the original coders. Second, even in earlier pilots, the high overlap of ideas in instances resulted in significant back-tracking during the coding process to ensure consistent ratings.

Another technique attempted was testing for statistical differences between conditions by randomly sampling instances from each condition and having judges identify an ordered relationship between those instances. For example, an instance from the 10-response condition and one from the 75-response condition would be compared by a judge with respect to originality, realisticness and quality. These judgments also had low IRR, which we attributed to the lack of ability to specify an equivalence relationship. The statistically significant differences identified did indicate that generally conditions requesting more responses resulted in more novelty, but the ordering was neither strong nor complete.

I think it is unlikely that there will be any near-term judge-free method for assessing outcomes other than novelty. However, novelty measures can be harnessed to drastically reduce the set of responses that have to be evaluated by judges. Beyond that, I think there

is potential for judging tasks that have individuals rank ideas for a metric rather than provide absolute scorings. An iterative cycle of these rankings in which the worst ideas are dropped at each step could provide a set of the "best" ideas more consistently than scale techniques.

7.7.1 Measurement with machine learning

In the course of this thesis, I regularly attempted to apply machine learning principles to reduce the reliance on judges. This may be the first concerted effort to do so in the context of disambiguating brainstorming responses. Unfortunately, these techniques were not successful enough to constitute meaningful contributions to this thesis.

The problem of entity resolution is well-documented. In the context of brainstorming, it turns out to be horribly expensive. Most entity resolution and clustering algorithms require a similarity metric between ideas. Several similarity metrics were attempted in the course of this work. The best, derived from traversal distance in WordNet between instances, gave only 0.16 Pearson correlation with a similarity metric derived from the judge-created idea forests. The natural response given the crowd-focused nature of this work is to appeal to the crowd for entity resolution. I tested this approach by implementing Wang et al.'s CrowdER [65], but found that performing a full disambiguation with the crowd was financially intractable. Furthermore, the results were not of a quality sufficient for this work.

Even with a perfect similarity metric, the value of clustering techniques for disambiguation is questionable. Brainstorming instances are overlapping in semantic content and come in multiple levels of generality. A perfect clustering would need a mechanism for enforcing a consistent level of generality between clusters, a problem so difficult even with human judges that it necessitated an explicit hierarchical structure in this work.

It seems the best course for automation moving forward is to focus on metrics that can skip the disambiguation step. Disambiguation is incredibly valuable, as in most of our datasets there were roughly half as many semantically meaningful ideas as their were instances. However, there are opportunities with novelty metrics in particular for automatic derivation and labeling. Some potential approaches are proposed as future work in Chapter 8.

7.8 Information and interaction — interventions

One core element of the task design in this work was to provide Osborne's rules of brainstorming. However, I am not convinced these rules had a significant impact on the quality of results. Specifically, not many ideas were combinations of other ideas in runs, and the principle of deferred evaluation may actually reduce overall quality. My intuition is that this is a result of the qualitative differences of brainstorming in a microtask environment: workers are being paid to complete the task. The first impact of this is that workers have no reason to be engaged with or motivated by the problem itself. The second impact is that deferred judgment actually incurs cost in this context. Since each idea must be paid for to guarantee production (this is evidenced by the low quantities when asking for any number of ideas, and the lack of use of the free-form section at the end of the task), every poor quality idea necessitates another idea solicitation. Explicitly condoning deferred judgment is thus at best uneconomical.

In addition, I must note the overwhelming positive response to the brainstorming HITs by workers. We regularly received feedback stating the task was fun — it may be that creative task HITs have an inherent competitive advantage against more mundane tasks such as labeling the transcribing. However, the workers would also regularly apologize for submitting HITs early and put significant effort into providing the remaining requested responses through email and other channels, despite the explicit statement in the task that reward would not be withheld for ethical reasons. I am concerned that this outpouring of goodwill was an explicit result of the task design, namely that the HIT stated a clear academic affiliation. It would be invaluable to understand exactly what impact academic affiliation has on result quality in this study and others.

Osborn's rules and academic affiliation make up the two divergent properties of the brainstorming task in this thesis from those commonly employed. However, there are many proposals for variants to brainstorming tasks which could improve results in one dimension or another. I propose the best approach for development is *interactive brainstorming*, brainstorming tasks that respond to the worker's input of ideas. For example, a task could identify a participant performing repeat riffing and propose an alternative strategy. While these interventions are an interesting component of the brainstorming task design space, I did not employ them explicitly in this work. Thus, I leave further speculation on potential manipulations of task design to future work in Chapter 8.

7.9 Theory of brainstorming

The most surprising finding of this work was that novelty, a proxy for quality, behaved in the opposite of the way expected. Rather than decrease over time as people expended their best ideas, novelty *increased* over time. In fact, it appears that brainstormers spend many of their first responses giving ideas that are from a common, "obvious" pool of ideas. In fact, over 50% of instances in the first 5 of a run are examples of the top 5% of ideas. It seems that asking for more responses actually pushes workers out of this comfort/common zone, at which point they branch off and explore different corners of the space of possible ideas. I liken this to the "burn-in" period before meaningful results common in sampling and iterative optimization techniques.

This exploration goes multiple ways. Rarely, there is the genuinely novel and thoughtful responder, who generates 75 or 100 high-quality ideas that make the rater exclaim "I want to implement this solution"! Often, the ideas take an element of the outrageous and unrealistic, though even absurd ideas contain an element of usefulness in inspiring orthogonal solutions. However, the most common strategy for generating ideas past the burn-in period seems to be changing the scope of the problem as described in Chapter 6.

There were also several unexpected behaviours in how workers chose to approach questions. Initially, task time was limited proportional to the number of responses requested. However, very quickly workers provided feedback that they wanted a significantly higher time allotment. It became clear that this was because many of the workers were accepting high-reward HITs for completion later. It is important to note that there is a culture of *hunting* for the highest-value tasks and claiming them for later completion. This became even more apparent in common responses to the turk question asking for mechanisms and policy changes to mitigate this behaviour and more evenly distribute high-reward work.

Workers would leave tasks open for long periods, causing spikes in the time to brainstorm ideas. Often the first ideas would take hours to be generated as workers claimed the HIT and left it open in the browser, or a spike would be witnessed partway through the task as workers took a break to do something else. It is unclear how these breaks impact response quality. Are participants stopping because they want to come back with a fresh mind and more ideas, or out of boredom? Furthermore, access to the Internet is a concern, and it may be that workers use these spikes in time to search for inspiration and information to provide further ideas.

Another artifact observed in the timing is how quickly workers were able to generate ideas between categories when compared to the time observed in prior work. Switching vs staying within a category had a difference of less than one second in this work, while prior

work cited a difference of 6-12 seconds. There are several possible reasons for this. It may be that workers on microtask marketplaces have demonstrably lower standards for their own responses, and thus will spend less time cognitively on ensuring the new direction is valuable. It may be that because workers choose to engage in a brainstorming task rather than being assigned it, they are likely to be more skilled at ideation.

The burn-in period and delayed fulfillment of HIT requirements both point towards a distinct theory of brainstorming in microtask marketplaces to that in traditional settings. These differences in how workers brainstorm almost certainly affect brainstorming outcomes. Thus, while this work set out to solve pragmatic problems of design, theoretical considerations will prove a necessary step in that pursuit.

7.10 Summary

In HCI and AI research, it can be commonly said that the goal is to provide mechanisms by which a user (in this case a requester) can provide the barest representation of their goal and the system will fulfill it. This thesis derives from these same aspirations applied to the domain of idea generation for crowd creativity. Fortunately for the continued employment of researchers (and this grad student), any such system requires solutions to a number of difficult problems as a prerequisite. This chapter has summarized intuitions as to solutions for these problems, as collected over a year and a half of exploration in the domain.

With regards to designing brainstorming prompts, it is likely that there exist design dimensions independent of widely diverging problem domains. Considerations such as provision of context can be applied to any question, but the possibility of "decorating" responses must be evaluated on a per-question basis. It seems that asking for as many ideas as possible and paying as much as possible seems to be the ideal, but it is inevitable that both of these values will eventually produce degenerating returns — the limits are just beyond those expected. I think it important to consider stopping criteria in brainstorming tasks more explicitly, and to explore the expanded domain of problems that can be solved by understanding the idea space. Evaluation criteria continue to present a phenomenal challenge, with judge-based techniques likely to remain the state of the art. Finally, the natural step forward for the design of brainstorming tasks is to introduce interventions. The most promising avenue is to harness the technological capability of crowd brainstorming platforms to produce *interactive brainstorming* systems, which respond to the worker's ideas and guide them to improve performance.

Chapter 8

Future work

8.1 Introduction

It is traditional to offer a brief future work section surveying the direct implications of research. However, as this is a foundational study, speculation on possibilities, outcomes and alternatives was inevitable. Thus, this chapter is dedicated to those research items I wanted to achieve, realized could be achieved, or wish I had achieved in the course of this thesis. If brainstorming is to be conducted effectively on microtask marketplaces, there is significantly more work to be done.

In particular, this chapter focuses on work that directly follows from the contributions of this thesis. Chapter 7 provides a more speculative discussion of the open problems in crowd brainstorming. The particular problems addressed in this chapter are:

- the expansive space of potential metrics for brainstorming
- opportunities for automating the judge-intensive processes employed in this work
- the need for additional models of brainstorming
- interventions and manipulations of brainstorming tasks with the goal of improving outcomes
- external validity and generalization

8.2 Metrics for brainstorming

This thesis has extracted metrics from brainstorming runs in two ways. The first is via idea disambiguation, and the resulting o-score metric of novelty. The second is via qualitatively-identified strategies. However, the hosts of ratings scales and metrics across scientific domains show that it is rarely sufficient to accept an initial measurement technique as fully descriptive of the phenomena of interest. This section explores a wider domain of potentially informative metrics, both within and without the labeled corpora provided by the idea forests.

8.2.1 Exploring the idea forest

The idea forest, unlike more traditional codings of brainstorming data, includes generalization in addition to disambiguation information. In this thesis, disambiguation was leveraged to provide a metric for novelty, but the idea forest structure has the potential to provide significantly more information when topology is accounted for.

In Chapter 3, the depth and breadth of category trees were discussed. The depth of a category tree provides an indication of the levels of detail at which a problem was explored. The depth of an individual idea node provides a relative measure of generality to its descendants and ancestors, and may even carry some meaning when compared against other nodes. The breadth of a tree can be measured in several ways, for example as the number of nodes in a tree, the mean number of hops between nodes, or the longest number of hops between nodes. Breadth measures provide an approximation for how divergently people consider a single category, and can be directly compared between category trees. Intuitively, depth and breadth measures are analogous to brainstorming outcomes of interest: specificity, and variety.

Novelty can be derived multiple ways given a labeled cluster forest. Another naive derivation would simply count an idea's novelty as its o-score calculated on a per-category rather than per-idea basis. More complex derivations may encode more information. For example, one could encode novelty as the sum of the idea masses of a node *and its descendants and ancestors*, weighted by their distance from the node of interest. This would encode the concept that an idea with very few instances may be less novel if it is a slightly more specific instantiation of a more general and popular idea. In contrast to the category o-score metric, it would not include siblings and cousins of the node in the score.

Any of these metrics may better describe the outcomes of interest in a brainstorming campaign. Furthermore, the outcomes of interest will no-doubt vary from application to

application. Idea forests were created in response to a need for disambiguation at scale, however it is clear that they encode a much more significant body of information that is worth leveraging.

8.2.2 Other encodings of brainstorming data

Idea forests were chosen for this work because they encoded generalization relationships, provided a solution for idea disambiguation, and could be created with a coding algorithm that was tractable for human judges at the scale enabled by microtask marketplaces. One limitation of ideas forests is their inability to encode multiple inheritance.

Consider the example instances "use the iPod to play music in the bathroom" and "use the iPod to listen to podcasts in the bathroom" (these instances are fabricated for purpose of example). It is clear the instances should have a common parent. However, it is unclear what the local structure of an idea forest would be in this case: would audio playback or the location of a bathroom provide a higher-level parent? Furthermore, it is likely that nodes for audio playback and the bathroom location would themselves have divergent parents.

A strictly bipartite graph of *dimensions* and instances could capture this notion. Essentially, every component of any instance that adds semantic meaning would be represented by a dimension, and an instance would be represented simply as a combination of those dimensions. This type of representation would provide new metrics of interest, such as the rate that new dimensions arise. Furthermore it would be possible to create a limited generative model of ideas, with new ideas constructed from dimensions similarly to the process of Latent Dirichlet Allocation for text documents. However, constructing such an encoding would once again come up against the limitations of machine learning techniques and human judges, with a necessity for the latter driven by the former and an inability to scale with thousands of responses.

8.2.3 Components of creativity

The choice of novelty for this thesis was a pragmatic one; capturing creativity as a whole proved a difficult and ill-defined task. As such, the models and findings presented have a limited applicability. To further brainstorming research in this environment, other components of creativity must be considered. As a brief survey, I re-state those discussed in Chapter 2: originality, feasibility, elaboration, and flexibility.

8.3 Automation

Generating this thesis required the substantial up-front cost of gathering and labeling a 10000-response corpus of brainstorming instances. While concepts such as novelty and uniqueness can be identified with this labeled set, the ideal brainstorming task-generator described in the introduction of this thesis would be able to automatically identify these and a spectrum of other metrics without a need for judge labeling. This section explores the possibilities for automatically arriving at outcomes of interest in brainstorming domains.

8.3.1 Finger-printing

One piece of relatively low-hanging fruit is to *fingerprint* workers. As shown in Chapter 5, workers brainstorm at different qualities. A fingerprint would encompass all the information that could be automatically detected about brainstorming style, such that it would be simple to identify that worker again in the future. For example, the prevalence of repeat riffing could be detected by Levenshtein edit distance, the reading comprehension level of the worker could be captured, their pattern of time spent to generate ideas, and so on.

One particularly important application for participant fingerprinting is to prevent gaming of the system. In the course of gathering the brainstorming corpus, several cases were identified in which two response sets from different worker IDs produced nearly identical instances to the same question. In another case, a participant provided identical instances as another — but the two HITs were not for the same question. This suggests that some Mechanical Turk workers are using multiple accounts to maximize revenue by performing the same task twice, in parallel. Fingerprints of workers could be used to detect this behaviour quickly and prompt a requester to investigate further.

8.3.2 Similarity without judges

In Chapter 4, I briefly discussed NLP-based methods for assigning similarity scores to brainstorming instances based on cosine similarity. In this section, I will describe other techniques for assigning similarity scores between instances.

Edit distance is a simple metric that can capture only superficial similarities between ideas. However, it can be expected to identify strategies such as repeat riffing with very high reliability. In the case of brainstorming responses, it could be valuable to consider edit distance at the character level or the word level.

The cosine similarity metric used in this thesis is based on bag of words representations that are stemmed and labeled with part-of-speech tags via WordNet. However, this is not the only way to encode semantic information. For example, work by Gabrilovich and Markovich has utilized the massive corpus of semantic information available in Wikipedia to produce representations of individual words as weighted vectors of their relatedness to articles [**?**]. This high-dimensional representation could be used to compute cosine similarity scores that capture more semantic information.

Another approach for computing similarity would be to consider topologies of related terms. For example, one instance could be compared to another by considering the mean distance between terms in those instances, as measured in hops through the WordNet graph (WordNet encodes relationships between words in a hierarchical topology not unlike an idea forest). Furthermore, this could be extended to the Wikipedia-based article representation above. Wikipedia can be considered as a topology of articles in which nodes are connected if there exist links between them. In this case, a similar number-of-hops metric could apply.

8.3.3 Automated construction of idea forests

Full automated construction of idea forests is likely an intractable problem in the near-term. However, it may be possible to approximate the metrics derived from an idea forest. In Chapter 4, correlation clustering was employed to produce an initial clustering of instances. In this case, the similarity metric employed was relatively weak (cosine similarity of bag of words) and the implementation of correlation clustering was extremely naive. A better implementation of correlation clustering and an advanced similarity metric (such as the Wikipedia-based semantic bag of words described above) could perform significantly better, especially if the target was to produce a flattened clustering of categories as opposed to individual ideas. This clustering could then be used as a disambiguation to provide an estimate of o-score.

8.3.4 Predicting novelty

A much more tractable approach is to directly approximate metrics such as novelty (alternatively, number of unique ideas or categories) rather than construct a disambiguation. In particularly, I expect novelty could be predicted with some accuracy by examining the similarity scores between instances in a run. Intuitively, a participant that produces a high o-score would have corresponding high similarity scores between instances.

The corpora in this thesis provide a candidate data set for exploring methods to make these predictions. A feature vector could be constructed that encoded summary statistics of each of the similarity metrics for the temporally-ordered instances in a run (edit distance, cosine similarity and topology-based for WordNet and Wikipedia), as well as a few easily-derived summary statistics of the run as a whole (reading level, number of words, etc). Then, machine learning techniques for regression such as linear regression or random forests could be applied to produce a prediction for the novelty of the participant.

This similarity-based approach to regression has several advantages. It does not require human judges. Furthermore, it is independent of the vocabulary of the problem domain, and the resulting regressor may even be applicable between-questions.

Such measures do not need a high granularity of accuracy to be useful. If, for example, a score could be computed between-questions with reasonable (better than random) accuracy, participants could submit a small batch of instances, have their novelty scores automatically computed by the regressor in real-time, and participants scoring in the upper quartile could be offered an opportunity to submit a much larger group of instances for a greater reward. This filtering would reduce the body of truly poor responses that requesters would need to filter through, and the threshold for offering further work could be adjusted according to budgetary requirements and the number of responses sought.

8.3.5 Active learning for presenting ideas

Once a body of ideas has been collected, it still needs to be examined by a human judge for evaluation and use further in the ideation process. Techniques from active learning could be applied here. For example, a simple k-means clustering approach could be used, where the requester labels a subset of randomly selected instances for idea or category, and then additional ideas for labeling are forwarded to the requester based on uncertainty. Again, a practical similarity metric would need to be employed. Requesters could continue to receive and label responses until a pool of satisfying responses had been collected, without the need to examine the entire data set. Using uncertainty to determine which responses to show the requester would increase the likelihood that the ideas received are novel (unseen by the requester so far). An active-learning approach would favour scenarios in which collection is cheap but there are limited human resources for filtering ideas later.

8.4 Models of brainstorming

This thesis presented models for quantity and novelty of ideas. Models of these outcomes were each in a specific context: brainstorming campaigns for quantity, and brainstorming runs for novelty. The design space of brainstorming outcomes and contexts includes far more models than could be reasonably explored in this thesis. This section will introduce a few models that could be particularly valuable.

8.4.1 Improved quantity models

The quantity models in this thesis have weaknesses. Most egregiously, the Bernoulli decay model with participant parameters estimates a decay curve for each participant based on a limited subsection of that curve. This is obviously problematic, and is a side effect of the decision to encode the temporal information regarding the order of worker contributions, when this temporal information should not impact each workers' individual contributions. An alternative model for quantity would generate categorical values for ideas from a different distribution for each worker. The following specification outlines the intent of such a model:

$cat_p \sim dirichlet(...)$

$idea_{pi} \sim categorical(cat_p)$

The relative value of participants could then be assessed a function of the posterior category probabilities.

8.4.2 Number of responses requested

In establishing the brainstorming corpora, participants were asked for varying numbers of responses. The impact of this condition was not examined in this thesis. Figure 8.1 illustrates the number of categories received as a function of the number of instances received over a brainstorming campaign, split by this condition. There is a near-strict increase of rate of idea generation as the number of responses requested increases.

This indicates that participants generating fewer ideas are likely to generate the *same* idea categories. The 75 condition, which bucks the increasing relationship, may also be an artifact of outlier participants, as multiple workers in this condition were identified as having unusually creative responses by researchers. Further modeling is needed to understand if this relationship is significant and if so, why it exists.

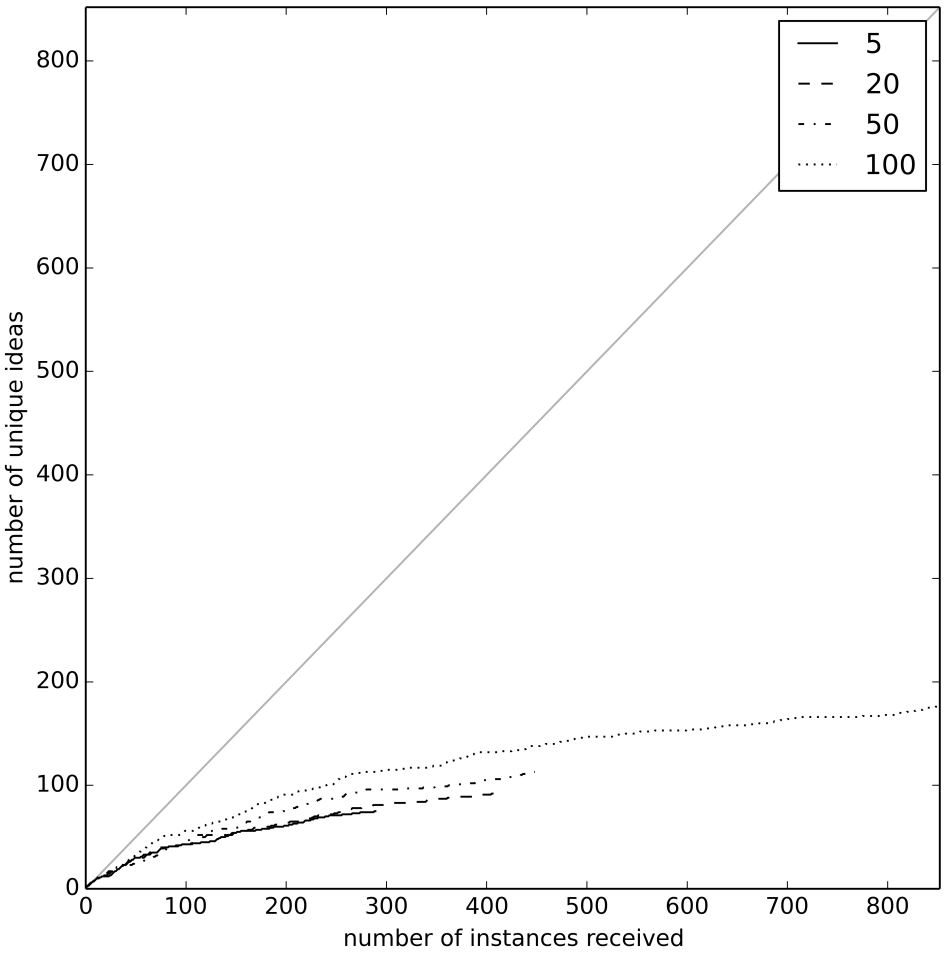


Figure 8.1: Categories over time for the iPod data

8.4.3 Dimensions of design

In Section 8.2.2, design dimensions were proposed as an alternative to idea forests. An interesting property of the rate of both idea and category generation is that they do not seem to converge; in Chapter 5 a positive minimum on the rate of idea generation was implied by the Bernoulli decay model fit. In contrast, it is conceivable that design dimensions would converge to a fixed or slower-growing set, as both ideas and categories can be conceived of as combinations of dimensions (with a naive count of 2^n combinations possible for n dimensions). It would also, then, be useful to model the relationship between design dimensions and ideas or categories, to understand whether new ideas and categories always necessarily encode new dimensions.

8.4.4 Saturation

In Chapter 5, the rate of new idea generation never decayed to zero; there were always more ideas to be generated. Intuitively, I had expected that it would be possible to reach a point where the idea pool had been *saturated*, and the probability of receiving a new idea was near zero.

In Chapter 6, a taxonomy of brainstorming strategies was given that implies qualitative differences between ideas. In particular, it would found that defocus scoped ideas formed the majority of the responses in the iPod question. It may be that these responses contribute dominantly to the non-zero growth lower bound. Intuitively, the number of theoretically possible ideas that are highly applicable to the specific question asked is much lower than the number possible ideas for defocus transformed question. A reasonable next step would be to model the rate of idea growth in the context of responses that do not employ defocus scoping, to determine if there is any saturation of the theoretical idea pool achieved in these corpora.

8.5 Interventions

The question I received most when describing this work to researchers was whether I had examined intervening in brainstorming sessions to change outcomes, as quantified by the models. With the baseline models now established, it is trivially possible for a researcher to manipulate the design of the brainstorming task presented in this thesis and compare outcomes for two of the most well-used metrics of brainstorming output: quantity of ideas

generated, and novelty. Throughout this work, I and my co-researchers proposed many such interventions, and I will briefly touch on them here.

First, it is useful to establish the goals of intervening. Throughout this thesis, I have often presented the ideal brainstorming outcome as a diagonal line representing a 1:1 growth relationship between ideas and instances. This corresponds to no "wasted" idea generation; each participant's work is directly contributing new information to the data set. One reasonable goal for intervention, then, is to close in one this growth rate. Another goal is to ensure high quality ideas. In this thesis, quality was considered only in terms of novelty, but one can also imagine intervening to produce ideas that are more realistic, more practical, and so on.

The following is a brief summary of interventions proposed in the course of this research.

- Feed participants seed responses from other brainstormers. This could increase growth rate by preventing overlap, or increase quality if ideas are improved upon.
- Detect poor performance in an on-line fashion. For example, detect repeat riffing with an edit distance metric. This could be used to filter users, or trigger a prompt to suggest alternative strategies.
- Stronger adherence to brainstorming guidelines. Isaksen [28] identifies several brainstorming requirements that are not met by the tasks in this work. For example, a training exercise could be provided to turkers and feedback given before the main brainstorming task, or ideation could be done in phases.
- Impose constraints on-line. If participants do veer towards defocus scoping, it could be practical to impose additional constraints in an on-line fashion to improve idea relevance. For example, in the iPod question, the participant could be asked to take into account a specific detail (i.e. music playback) for each batch of 10 responses.
- Microtask marketplaces such as Mechanical Turk provide functionality for filtering participants based on criteria. This could be used to assign credentials that enforce sufficient expertise for brainstorming problems.
- Dow et al. [17] found that crowd workers performed better when asked to rate their own work on a rubric. This intervention could easily be tested in the context of an ideation task.
- Participants could be asked to provide category labels for their own responses, to identify relationships. This could influence the ideation process, but also opens up the possibilities for on-line reactions to quantity metrics.

Ultimately, I find it is as possible to generate hundreds of ideas for brainstorming interventions as it was for my participants to generate hundreds of uses for an old iPod. However, it is my belief that *interactive* interventions, or those which respond to the user's input, would be most effective in raising the rate of production to the ideal. Interactive brainstorming interventions allow the system to directly request the attention of the worker. This is valuable, given that many workers will skip over instructions in a task. For example, workers sent messages to the researchers in the course of the study asking that their work not be rejected (and thus they would not receive payment) despite the HIT explicitly stating as part of the ethics guidelines that all work would be accepted. Interactive brainstorming interventions include prompts to change strategies, imposing constraints, and self-rating or self-labeling by workers.

8.6 Generalization

The goals of any foundational work on a topic must include external validity. In this thesis, four brainstorming problems were considered to help achieve this goal, but this is of course insufficient. In this section, future work to aid the generalizability of this work is discussed.

8.6.1 Question domains

All four questions in this thesis asked participants to provide textual ideas to solve problems. However, the domain of idea generation and brainstorming tasks is much larger. For example, the chair designing tasks of Yu and Nickerson [69]. A natural next step for this work would be to consider these alternative domains of ideation and examine the applicability of the proposed methods and models.

8.6.2 Understanding questions

In Chapter 5 it was shown that the rate of idea generation is different depending on the brainstorming question asked. This makes intuitive sense, but it is difficult to understand what exactly about questions contributes to these differences. In particular, it is useful to know if there are properties of brainstorming questions that affect the qualities of responses received to those questions.

Three possible examples of properties that may have an impact are: how constrained the question is; the degree of expertise in the question domain that is expected from the

participants; and examples given. Ideally, it would be possible to predict *a priori* of data capture some of the outcomes of these properties, such that questions could be adjusted to better achieve desired outcomes.

8.6.3 Comparison to traditional brainstorming

Chapter 5 explored a limited link between normal group and crowd brainstorming by replicating the predictions of the SIAM model. In this case, the properties of the two environments seemed alike. However, in Chapter 2, both group and electronic brainstorming settings were seen to encode their own explicit blocking and facilitating effects on brainstorming performance.

Brainstorming in microtask marketplaces is qualitatively different from traditional brainstorming:

- participants are spatially and temporally separated
- they accept HITs with a delay before participating
- they may take long breaks in the process of the brainstorming task
- they choose tasks rather than being assigned to them
- they may be filtered by qualifications
- they have access to the Internet

It is likely that these differences create their own blocking and facilitating effects. Understanding and modeling these effects is critical to producing strong ideation environments for microtask marketplaces.

8.7 Summary

This chapter enumerated the avenues for future exploration of brainstorming in microtask marketplaces. It focused on topics related to the primary goals of this thesis: to establish models and methods. In brief, this chapter suggests the development of additional metrics for brainstorming to be explored in a quantitative manor; the automatic extraction of

brainstorming metrics using NLP and ML techniques; the creation of additional models of brainstorming; interventions to improve the outcomes of brainstorming by altering task design; and finally, the additional work necessary to establish the generalizability of the claims in this thesis.

Chapter 9

Conclusion

Microtask marketplaces are a natural fit to creative tasks, enabling their automation in ways that have been until recently impossible. Existing work engaging crowds in creative tasks has taken the crowd's creativity for granted, and skipped over questions as to the nature and effectiveness of crowd workers in this problem domain. There are many open questions when it comes to the design of brainstorming tasks:

- What is the design space of a brainstorming task?
- What is the design space of a brainstorming *prompt* (the question to which the worker must provide answers)?
- How many ideas should be requested from each worker?
- How many workers should be be asked for ideas?
- Who should be asked? How can appropriate brainstormers be identified?
- How much should workers be paid for this kind of task?
- What is the stopping criteria for gathering responses? Can this be evaluated automatically?
- What is the evaluation criteria for responses? Can this be evaluated automatically?
- What information should the task expose to the worker?
- Should the task respond to the worker? If so, how?

- How do workers brainstorm differently in microtask marketplaces than other environments?

This work was a direct response to the lack of satisfying measurement techniques, models, and guidelines of brainstorming with which work could be interpreted and compared. I addressed this open research area in three parts. First, *idea forests* were introduced as a methodology and evaluation criteria for brainstorming responses at a scale typical of crowdsourcing marketplaces, where hundreds of participants can be employed. Second, quantitative models of idea novelty were constructed to make inferences as to the rate at which new ideas were generated, how individuals affected this rate, and when individuals generated their most novel ideas. These properties have direct implications for determining the worker and request counts to design for. Third, I developed a taxonomy of *strategies* employed by brainstormers in crowd contexts through a qualitative examination of the brainstorming corpus, which is a first examination of how workers brainstorming differently in crowd environments. In this chapter, each of these contributions will be reviewed.

9.1 Idea forests for quantifying brainstorming output

This thesis introduced *idea forests*, a hierarchical representation of a brainstorming corpus which encodes generalization relationships between responses and allows for disambiguation of ideas and as a result, measures for quantity and novelty. Traditional methods for labeling brainstorming data are intractable at the scales enabled by microtask marketplaces. The construction process for idea forests mitigates this using a tree-traversal algorithm for constructing idea forests which allows localized decision-making. A 10000-response brainstorming corpus across four questions was constructed and encoded as an idea forest, and will be made publicly available as a further contribution of this work. Further combating the problems of labeling at scale, a simulation-based method for testing the validity of claims based on idea forests is given, which can be performed without a complete parallel coding of a data-set.

The primary contributions of the idea forest structure were to enable the models and conclusions presented in the remainder of the thesis, and to refine the mechanisms of this research such that they are available to and repeatable by the research community.

9.2 The quantitative modeling of brainstorming

This thesis presented models of three properties of crowd brainstorming: the rate at which new ideas are generated; the novelty of ideas within a single participant's brainstorming run; and the properties of semantic category changes in a brainstorming run, as originally tested by Nijstad and Stroebe [47]. By fitting these models to the brainstorming corpus, several conclusions are drawn:

1. The rate of idea generation is non-linear, and subject to exponential decay.
2. Individuals are a significant source of variation in the quantity of unique ideas generated, with productive participants producing dozens more.
3. The novelty of generated ideas increases as participants ideate, reaching a peak after their 18th instance.
4. Participants are more likely to generate subsequent ideas within the same semantic category than expected by random chance.
5. It takes longer for participants to generate ideas when switching between semantic categories.

While these findings are useful, the more valuable contributions of this research are the models used to derive them, which are the first to be applied in this domain to empirically derive properties such as an individual's ability or novelty of ideas over the course of a brainstorming run. These models can be applied in future crowd brainstorming work to describe the statistical impact of interventions or be applied in a commercial setting to assess performance.

9.3 Qualitative strategies of idea generation

This thesis presented a taxonomy of brainstorming strategies that participants employed to generated ideas, derived from patterns observed in the corpus. These strategies describe how participants brainstorm in a microtask marketplace environment, and inform potential future interventions to improve brainstorming task design and processing. Participants would re-scope the problem to answer transformed questions, riff on old answers by keeping some element constant, and provide partial solutions which required further ideation to

resolve the original prompt. Finally, the chapter described potential applications of the strategy taxonomy, including automatic detection of certain trends, filtering responses, and proactively prompting users to engage in strategies that are particularly productive.

APPENDICES

Appendix A

Stan specification for models

A.1 Exponential decay model

```
1
2  data {
3      int N; // number of instances
4      real y[N]; // the number of ideas or categories receieved up to and including instance
           n
5      int x[N]; // ordinal position of the instance in its condition
6  }
7
8  parameters {
9      real<lower=0, upper=1> rate;
10     real<lower=0> y_scale;
11     real<lower=0> sigma;
12 }
13
14 model {
15     real mu[N];
16     for (i in 1:N) {
17         mu[i] <- y_scale * pow(x[i], rate);
18         y[i] ~ normal(mu[i], sigma);
19     }
20 }
```

A.2 Decaying Bernoulli model

```
1
2  data {
3      int <lower=0> N; // number of instances
4      int <lower=0, upper=1> novel[N]; // whether there was a novel idea at this point
5      int <lower=0> x[N]; // ordinal position of the instance in its condition
6  }
7
8  parameters {
9      real <lower=-100, upper=0> rate;
10     real <lower=0, upper=1> min_rate;
11 }
12
13 model {
14     real theta;
15     for (i in 1:N) {
16         theta <- min_rate + exp(rate * x[i]) * (1-min_rate);
17         novel[i] ~ bernoulli(theta);
18         //increment_log_prob(bernoulli_log(novel[i], theta));
19     }
20 }
```

A.3 Decaying Bernoulli model with participant parameters

```
1
2  data {
3      int <lower=1> M; // number of participants
4      int <lower=M> N; // number of instances
5      int <lower=0, upper=1> novel[N]; // whether there was a novel idea at this point
6      int <lower=0> x[N]; // ordinal position of the instance in its condition
7      int <lower=1, upper=M> participant[N]; // which participant provided response
8  }
9
10 parameters {
11     real <lower=-10, upper=0> rate[M];
12     real <lower=0, upper=1> min_rate;
13
14     real <lower=-10, upper=0> hyper_rate_mu;
15     real <lower=0, upper=5> hyper_rate_sigma;
16 }
17
18 model {
19     real theta;
20
21     for (i in 1:M) {
22         rate[i] ~ normal(hyper_rate_mu, hyper_rate_sigma);
23     }
24
25     for (i in 1:N) {
26         theta <- min_rate + exp(rate[participant[i]] * x[i]) * (1-min_rate);
27         increment_log_prob(bernoulli_log(novel[i], theta));
28     }
29 }
```

A.4 Comparison model of exponential decay and decaying Bernoulli

```
1
2  data {
3      // shared data
4      int N; // number of instances
5      int x[N]; // ordinal position of the instance in its condition
6
7      // bernoulli outcome variable
8      int novel[N]; // whether there was a novel idea at this point
9
10     // exponential outcome variable
11     real y[N]; // the number of ideas or categories receieved up to and including instance
           n
12 }
13
14 parameters {
15     // bernoulli parameters
16     real <lower=-10, upper=0> b_rate;
17     real <lower=0, upper=1> b_min_rate;
18
19     // exponential parameters
20     real<lower=0, upper=1> e_rate;
21     real<lower=0, upper=2> e_y_scale;
22     real<lower=0, upper=N> e_sigma;
23
24     // mixture parameter
25     real<lower=0, upper=1> lambda;
26 }
27
28 model {
29     // prior on lambda emphasizing even mix
30     lambda ~ beta(1, 1);
31
32     for (i in 1:N) {
33         real b_theta;
34         real e_mu;
35
36         real b_lp;
37         real e_lp;
38
39         b_theta <- b_min_rate + exp(b_rate * x[i]) * (1 - b_min_rate);
40         b_lp <- bernoulli_log(novel[i], b_theta);
41
42         e_mu  <- e_y_scale * pow(x[i], e_rate);
43         e_lp <- normal_log(y[i], e_mu, e_sigma);
44
45         increment_log_prob(log_sum_exp(log(lambda) + b_lp, log1m(lambda) + e_lp));
46     }
47 }
```

A.5 Novelty within brainstorming run model

```
1
2  data {
3      int<lower=0> N; // number of instances
4      int<lower=1, upper=100> order[N]; // order of instance in brainstorming run
5      real<lower=0, upper=1> oscore[N]; // oscore of instance at position
6  }
7
8  parameters {
9      real<lower=1, upper=100> split;
10
11     real<lower=0,upper=1> phi1;
12     real<lower=0.1> lambda1;
13     real<lower=0,upper=1> phi2;
14     real<lower=0.1> lambda2;
15
16 }
17
18 transformed parameters {
19     real<lower=0> alpha1;
20     real<lower=0> beta1;
21
22     real<lower=0> alpha2;
23     real<lower=0> beta2;
24
25     alpha1 <- lambda1 * phi1;
26     beta1 <- lambda1 * (1 - phi1);
27
28     alpha2 <- lambda2 * phi2;
29     beta2 <- lambda2 * (1 - phi2);
30 }
31
32 model {
33     real mix[N];
34
35     phi1 ~ beta(1,1);
36     lambda1 ~ pareto(0.1,1.5);
37     phi2 ~ beta(1,1);
38     lambda2 ~ pareto(0.1,1.5);
39
40     for (i in 1:N) {
41         if (order[i] <= split)
42             mix[i] <- order[i] / split;
43         else
44             mix[i] <- 1;
45
46         oscore[i] ~ beta((1 - mix[i]) * alpha1 + mix[i] * alpha2,
47                          (1 - mix[i]) * beta1 + mix[i] * beta2);
48     }
49
50 }
```

A.6 Idea generation time model

```
1
2  data {
3      int N;
4      int y[N];
5  }
6
7  parameters {
8      real<lower=1> mu;
9      real<lower=1> sigma;
10 }
11
12 model {
13     for (i in 1:N) {
14         y[i] ~ lognormal(mu, sigma);
15     }
16 }
```

References

[1] Amazon mechanical turk - welcome.

[2] Teresa Amabile and Teresa M. Amabile. *The social psychology of creativity*, volume 11. Springer-Verlag New York, 1983.

[3] Shai Bagon and Meirav Galun. Large scale correlation clustering optimization. arXiv e-print 1112.2903, December 2011.

[4] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89113, 2004.

[5] Barry L. Bayus. Crowdsourcing new product ideas over time: an analysis of the dell IdeaStorm community. *Management Science*, 59(1):226244, 2013.

[6] Michael S. Bernstein, Greg Little, Robert C. Miller, Bjrn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. Soylent: A word processor with a crowd inside. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, page 313322, New York, NY, USA, 2010. ACM.

[7] Thomas J. Bouchard Jr. Personality, problem-solving procedure, and performance in small groups. *J Appl Psychol*, 1969.

[8] Thomas J. Bouchard Jr and Melana Hare. Size, performance, and potential in brainstorming groups. *Journal of applied Psychology*, 54(1p1):51, 1970.

[9] Robert O. Briggs, Bruce A. Reinig, Morgan M. Shepherd, Jerome Yen, and J. F. Nunameker. Quality as a function of quantity in electronic brainstorming. In *System Sciences, 1997, Proceedings of the Thirtieth Hawaii International Conference on*, volume 2, page 94103. IEEE, 1997.

[10] Vincent Brown and Paul B. Paulus. A simple dynamic model of social factors in group brainstorming. *Small Group Research*, 27(1):91114, 1996.

[11] L. Mabel Camacho and Paul B. Paulus. The role of social anxiousness in group brainstorming. *Journal of Personality and Social Psychology*, 68(6):1071, 1995.

[12] Panayiota A. Collaros and Lynn R. Anderson. Effect of perceived expertness upon creativity of members of brainstorming groups. *Journal of Applied Psychology*, 53(2p1):159, 1969.

[13] Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, page 1318, 2005.

[14] G. Demartini, D. E. Difallah, and P. Cudr-Mauroux. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, page 469478, 2012.

[15] Alan R. Dennis and Joseph S. Valacich. Group, sub-group, and nominal group idea generation: new rules for a new media? *Journal of Management*, 20(4):723736, 1994.

[16] Michael Diehl and Wolfgang Stroebe. Productivity loss in brainstorming groups: Toward the solution of a riddle. *Journal of personality and social psychology*, 53(3):497509, 1987.

[17] Steven Dow, Anand Kulkarni, Scott Klemmer, and Bjrn Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, CSCW '12, page 10131022, New York, NY, USA, 2012. ACM.

[18] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(1):116, 2007.

[19] Ronald A. Finke, Thomas B. Ward, and Steven M. Smith. *Creative cognition: Theory, research, and applications*. MIT press Cambridge, MA, 1992.

[20] Evgeniy Gabrilovich and Shaul Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34(2):443, 2009.

[21] William Gale and Geoffrey Sampson. Good-turing smoothing without tears. *Journal of Quantitative Linguistics*, 2(3):217237, 1995.

[22] R. Brent Gallupe, Alan R. Dennis, William H. Cooper, Joseph S. Valacich, Lana M. Bastianutti, and Jay F. Nunamaker. Electronic brainstorming and group size. *Academy of Management Journal*, 35(2):350369, 1992.

[23] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian data analysis*. CRC press, 2013.

[24] Andrew Gelman and Cosma Rohilla Shalizi. Philosophy and the practice of bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66(1):838, 2013.

[25] J.R. Gibb. The effect of group size and of threat reduction upon creativity in a problem solving situation. *American Psychologist*, 6(7):324, 1951.

[26] William JJ Gordon. Operational approach to creativity. *Harvard Business Review*, 34(6):4151, 1956.

[27] Hassan Ait Haddou, Guy Camilleri, and Pascale Zarat. Dynamic models for ideas number prediction in brainstorming. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, page 382391, 2012.

[28] Scott G. Isaksen. *A review of brainstorming research: Six critical issues for inquiry*. Creative Research Unit, Creative Problem Solving Group-Buffalo, 1998.

[29] David G. Jansson and Steven M. Smith. Design fixation. *Design Studies*, 12(1):311, 1991.

[30] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[31] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut. Crowdforge: Crowdsourcing complex work. In *Proc. of UIST*, page 4352, 2011.

[32] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 453456, New York, NY, USA, 2008. ACM.

[33] Aaron Michael Koblin. The sheep market. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, C&C '09, page 451452, New York, NY, USA, 2009. ACM.

[34] John Kruschke. *Doing Bayesian Data Analysis: A Tutorial Introduction with R.* Academic Press, 2010.

[35] A. P. Kulkarni, M. Can, and B. Hartmann. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *CHI extended abstracts*, page 20532058, 2011.

[36] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[37] Sheena Lewis, Mira Dontcheva, and Elizabeth Gerber. Affective computational priming and creativity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 735744. ACM, 2011.

[38] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Turkit: tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, page 2930, 2009.

[39] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Turkit: human computation algorithms on mechanical turk. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, page 5766, 2010.

[40] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD workshop on human computation*, page 6876, 2010.

[41] Irving Lorge, Jacob Tuckman, Louis Aikman, Joseph Spiegel, and Gilda Moss. SOLUTIONS BY TEAMS AND BY INDIVIDUALS TO a FIELD PROBLEM AT DIFFERENT LEVELS OF REALITY. *Journal of Educational Psychology*, 46(1):17, 1955.

[42] Richard L. Marsh, Joshua D. Landau, and Jason L. Hicks. How examples may (and may not) constrain creativity. *Memory & Cognition*, 24(5):669680, 1996.

[43] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, page 775780, 2006.

[44] George A. Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):3941, 1995.

[45] Brian Mullen, Craig Johnson, and Eduardo Salas. Productivity loss in brainstorming groups: A meta-analytic integration. *Basic and applied social psychology*, 12(1):323, 1991.

[46] Jeffrey V. Nickerson and Yasuaki Sakamoto. Crowdsourcing creativity: Combining ideas in networks. In *Workshop on Information in Networks*, 2010.

[47] Bernard A. Nijstad and Wolfgang Stroebe. How the group affects the mind: A cognitive model of idea generation in groups. *Personality and social psychology review*, 10(3):186213, 2006.

[48] Jon Noronha, Eric Hysen, Haoqi Zhang, and Krzysztof Z. Gajos. Platemate: crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, page 112, 2011.

[49] Alex F. Osborn. Applied imagination: Principles and procedures of creative problem-solving. 1957. *New York: Charles Scribners Sons*, 1957.

[50] Morris B. Parloff and Joseph H. Handlon. The influence of criticalness on creative problem-solving in dyads. *Psychiatry: Journal for the Study of Interpersonal Processes*, 1964.

[51] Sidney J. Parnes. Effects of extended effort in creative problem solving. *Journal of Educational Psychology*, 52(3):117–122, June 1961.

[52] Sidney J. Parnes and Arnold Meadow. Effects of" brainstorming" instructions on creative problem solving by trained and untrained subjects. *Journal of Educational Psychology*, 50(4):171, 1959.

[53] Paul B. Paulus, Vicky L. Putman, Karen Leggett Dugosh, Mary T. Dzindolet, and Hamit Coskun. Social and cognitive influences in group brainstorming: Predicting production gains and losses. *European review of social psychology*, 12(1):299325, 2002.

[54] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. WordNet:: similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, page 3841, 2004.

[55] Alain Pinsonneault, Henri Barki, R. Brent Gallupe, and Norberto Hoppen. Electronic brainstorming: The illusion of productivity. *Information Systems Research*, 10(2):110133, 1999.

[56] Daniel L. Roenker, Charles P. Thompson, and Sam C. Brown. Comparison of measures for the estimation of clustering in free recall. *Psychological Bulletin*, 76(1):45, 1971.

[57] Marie Christine Roy, Stephane Gauvin, and Moez Limayem. Electronic group brainstorming the role of feedback on productivity. *Small Group Research*, 27(2):215247, 1996.

[58] Mehran Sahami and Timothy D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, page 377386, 2006.

[59] Jami J. Shah, Steve M. Smith, and Noe Vargas-Hernandez. Metrics for measuring ideation effectiveness. *Design studies*, 24(2):111134, 2003.

[60] Arina Soukhoroukova, Martin Spann, and Bernd Skiera. Creating and evaluating new product ideas with idea markets. *Produktinnovation mit*, page 94, 2007.

[61] Stan Development Team. Stan: A c++ library for probability and sampling, version 2.2, 2014.

[62] Donald W. Taylor, Paul C. Berry, and Clifford H. Block. Does group participation when using brainstorming facilitate or inhibit creative thinking? *Administrative Science Quarterly*, 3(1):23–47, June 1958. ArticleType: research-article / Full publication date: Jun., 1958 / Copyright 1958 Johnson Graduate School of Management, Cornell University.

[63] Joseph S. Valacich, Alan R. Dennis, and Jay F. Nunamaker. Group size and anonymity effects on computer-mediated idea generation. *Small Group Research*, 23(1):4973, 1992.

[64] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, page 319326, New York, NY, USA, 2004. ACM.

[65] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. CrowdER: crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):14831494, 2012.

[66] Warren E. Watson, Larry K. Michaelsen, and Walt Sharp. Member competence, group interaction, and group decision making: A longitudinal study. *Journal of Applied Psychology*, 76(6):803, 1991.

[67] S. E. Whang, J. McAuley, and H. Garcia-Molina. Compare me maybe: Crowd entity resolution interfaces. 2012.

[68] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Strategies for crowdsourcing social data analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 227236, New York, NY, USA, 2012. ACM.

[69] Lixiu Yu and Jeffrey V. Nickerson. Cooks or cobblers?: crowd creativity through combination. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, page 13931402, 2011.

[70] Lixiu Yu and Jeffrey V. Nickerson. An internet-scale idea generation system. *ACM Trans. Interact. Intell. Syst.*, 3(1):2:12:24, April 2013.

[71] Salvatore V. Zagona, Joe E. Willis, and William J. MacKinnon. Group effectiveness in creative problem-solving tasks: An examination of relevant variables. *The Journal of Psychology*, 62(1):111137, 1966.

[72] H. Zhang, E. Law, R. Miller, K. Gajos, D. Parkes, and E. Horvitz. Human computation tasks with global constraints. In *Proc. of CHI*, page 217226, 2012.